MAGISTERARBEIT

(Mag. rer. soc. oec.)

Design und Implementierung eines Internet basierten Content Management Systems

Ausgeführt am Institut für
Scientific Computing
der Universität Wien

unter der Anleitung von
Ao.Univ.Prof. Dipl.-Ing. Dr. Siegfried Benkner

durch **Thomas Gerhardt**
Resedaweg 65
A-1220 Wien

Wien, am 14. Juni 2007

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Wien, 14. Juni 2007 Thomas Gerhardt

0. Einleitung:

Die nachfolgende Arbeit ist aus den an mich gestellten Anforderungen für die Realisierung eines Content Management Systemes entstanden. Ich war als IT Manager bei einem Automobilimporteur beschäftigt. Aus Gründen der Konkurrenz unter den Unternehmen wurde diese Arbeit weitgehend anonymisiert dargestellt. Das Ergebnis meiner Arbeit war es, aus einem aus statischen Webseiten bestehendem Environment ein dynamisches Content Management System zu implementieren.

In der vorliegenden Arbeit soll dargestellt werden, wie die Vorgehensweise war um aus der bestehenden Landschaft von Betriebssystem, Hardware und Software eine neue Umgebung zu schaffen.

Die Unternehmensstrategie war es die bestehenden Web Seiten in ein CMS System umzubauen. Die Entscheidung für die Wahl es CMS hängt ab von einer Reihe von Kriterien, die vom Unternehmen selbst bestimmt werden. Die wichtigsten zeigen sich im direkten Vergleich. CMSse gibt es wie Sand am Meer, allein in Europa rund 500. Somit kann die Auswahl eines CMS ein langes und nicht gerade einfaches Unterfangen werden. Der erste Schritt der Evaluierung besteht darin, die Systeme nach Ihren Möglichkeiten zu bewerten. Es wurden von mir 3 Systeme in die engere Wahl genommen und ich entschied mich dann für die Zope Umgebung. Es konnte mit diesem System die größte Flexibilität erreicht werden und die Wünsche unseres Marketings konnten perfekt umgesetzt werden.

Es wurden Analysen von Anforderungen gemacht mit einem Schwerpunkt auf das Backend der Software um es Usern zur Datenpflege zur Verfügung zu stellen. Eine dynamische Einpflege der Daten (durch tägliches Update der Daten durch eine dritte Quelle) war ebenso wichtig in der Umsetzung wie ein performanter Userfreundlicher Auftritt im Internet. Es wird hier der Ansatz gezeigt, wie einerseits die User die Daten einpflegen können, um die Inhalte der Internetseite anzupassen. Ich erkläre auch die Einbindung eines Newsletters in unsere Seite, der wichtige Daten für das Marketing liefert.

Ein weiterer Teil der Arbeit ist es die täglich ankommenden Daten aus der Konzernzentrale zu verarbeiten. Hier werden Text Dateien aufbereitet, kontrolliert und für den Car Konfigurator auf der Web Seite in eine MySql Datenbank eingestellt.

In weiteren Abschnitten werden die richtige Skalierung des Systems und die richtige Anbindung der Serverumgebung dargestellt welche durch die Software Tests mit einem Load Test Tool festgestellt wurden. Das Load Testing Tool der Firma Mercury war hier das Kerninstrument für diese Analysen. Mit diesem Tool wurde das noch nicht life gestellte neue System auf Last getestet. Man konnte in diesem Tool so genannte Virtuelle User definieren, die einen vorgegebenen Ablauf auf der Internetseite durchlaufen. Während dieses Durchlaufes wurden laufend Messdaten erhoben um einen Schluss auf die Performance der Internetseite zuzulassen. Diese Tests werden anhand von Grafiken und Scripten erklärt.

Die mitgelieferten Linux Tools rundeten die Informationen für das System noch ab. Nagios wurde herangezogen um diese Linux Tools grafisch darzustellen und tägliche Auswertungen über Bandbreite, Zugriffe, etc. zu liefern. Die Ergebnisse und Anforderungen wurden ebenso in dieser Arbeit dokumentiert.

Abschließend werden noch Backup und Wartung des Systems mittels cold Standby System und System Monitoring mittels System Überwachung Tools erörtert. Hier werden das Backup der aktuellen Daten und der Datenbanken beschrieben. Die Übertragung der Daten auf ein cold Standby System sollen zusätzlichen Ausfallschutz bieten. Das Laufende Monitoring mit einem Linux Tool sollen aktuelle Probleme mit einfacher Email Alarmierung anzeigen.

0.1. KURZFASSUNG:

Die Anforderung war es aus einem statischen Modell von html Seiten ein dynamisches Content Management zu machen.
Es wurde einerseits ein neues Betriebssystem ausgewählt, aber auch, unter dem Aspekt eine freie Software einzusetzen, ein geeignetes Content Management System.
Die an mich gestellten Anforderungen waren sehr vielschichtig, da einerseits IT Software Architekten damit arbeiten sollten, aber auch Enduser, die die eigentliche tägliche Datenpflege übernehmen sollten. Eine externe Schnittstelle zur Datenversorgung sollte ebenso mit eingebunden werden.
Dies alles sollte in einer neuen homogenen Landschaft von Server, Backupserver und Mailserver integriert werden.
Für die neue Umgebung wurde ein Tool zur Performance der neuen Umgebung herangezogen. Das Tool sollte mir durch Simulation von virtuellen Usern, die Performance und die Systemschwachstellen vor dem Onlinebetrieb liefern.
Eigens implementierte Testszenarien für die Messung der einzelnen Ebenen der neuen Plattform wurden mit diesem Tool simuliert. Auf diese Auswertungen und Analysen wurden großer Wert gelegt, daher auch sehr detaillierte Ausführungen in dieser Arbeit.
Ein Backupkonzept für die tägliche und monatliche Sicherung und die Implementierung von Monitoring Tools, die mit Grafiken der Bandbreiten und einigen Grafiken aus der Analyse dokumentiert werden, sorgten für weitere Diskussionspunkte in der nachfolgenden Arbeit.
Letztlich wird die Onlinestellung und der Echtbetrieb dieser neuen Umgebung beschrieben.

1. Anforderungserfassung:

1.1. Kurzbeschreibung der potentiellen Anforderungen

1.1.1. Ausgangssituation

Die momentane Ausgangssituation erforderte es, von unserem starren System der Internetseiten auf ein flexibles CMS System umzustellen. Unser derzeitiger Internetauftritt umfasst ein System von 100 statischen html Seiten, die lediglich durch Links miteinander verbunden sind. Die momentane Internetpräsenz des Konzerns basiert auf dem intentionalen Zugangsmodell. Die Struktur ergibt sich nicht durch eine klassische Zielgruppen-Segmentierung, auch nicht durch einen Zugang nach Produktkategorien, sondern durch die angenommenen Intentionen der Nutzer. Das intentionale Modell muss aber dennoch unter Rücksichtnahme auf Zielgruppen konzipiert werden, und vor allem die Erfüllung der wesentlichen Funktionen eines solchen Portals sicherstellen. Die Funktionen können im Wesentlichen wie folgt definiert werden:

- Presales: Kernfunktion des Portals, Informationen für Nutzergruppen mit anstehender oder bereits getroffener Kaufentscheidung
- Postsales: Nachbetreuung bestehender Kunden im Sinne von Kundenbindung
- Imaging, Markenkommunikation

Der aktuelle Wartungsaufwand ist teilweise unabsehbar, denn es kann vorkommen, dass eine Information an mehreren Stellen platziert werden muss und dies eine langwierige Suche im html Code erforderlich wird
Im zweiten Schritt soll auch das Layout an bestehende Vorgaben angepasst werden. Im Detail soll eine klare Darstellung der einzelnen Modelle vorgenommen werden. Die weiteren Bereiche der Homepage sollen klar gegliedert werden und ebenfalls in klarer übersichtlicher Form teils in eigenen Navigationspunkten, teils im Content selbst eingebunden werden Im letzten Schritt soll das interne Personal in der Lage sein Eingaben von neuen Modellen, News und kleine Änderungen selbständig durchzuführen und dadurch effizienter zu Arbeiten und letztlich auch eine Kostenersparnis zu erzielen.

Die Anforderungen können durch die derzeitige Präsenz nur zum Teil erfüllt werden. Diese Tatsache ist in erster Linie durch die Inhomogenität der Weblandschaft zu erklären:

- Zahlreiche, verschiedene URL´s
- Keine übergeordnete Plattformstruktur, zahlreiche Unterseiten (eigene Fenster, tlw. abweichendes Design)
- Contentlücken, teilweise unzureichende Produktdarstellung, tlw. zu wenig Informationen für bestehende Kunden.
- Funktionsumfang, kein Konfigurationstool verfügbar

Diese Vorgaben erforderten diverse Entscheidungen.
1.) auf ein Content Management System aufbauen, welches mir genug Flexibilität und Möglichkeiten bot die gesetzten Ziele zu erreichen.
2.) eine Datenbankanbindung schaffen, um die Informationen künftig effizienter bearbeiten zu können.
3.) einen Webserver auswählen, eine geeignete Datenbanken und ein CMS System finden ohne hohen Kostenaufwand.
Folgende konzeptionelle Ansätze sollten realisiert werden:

URL Problematik, Zugänge	→	Vereinheitlichung bzw. Distributionsportal, Zugang zu allen Bereichen (Bank, Nutzfahrzeuge, etc..)
Contentlücken	→	Ergänzung der Inhalte, Vereinheitlichung des Niveaus über die gesamte Modellpalette
Designabweichungen	→	Designseitige Überarbeitung auf Basis der Vorgaben (Farb-. Formwelt), Templates
POP up Problematik	→	Steigerung des inhaltlichen Integrationsgrades, Reduktion der Anzahl erforderlicher Browserfenster
Funktionsumfang	→	Erweiterung im Sinne der Presales-Funktion (Konfigurator)

1.1.2. Externe Sicht eines Systems

1.1.2.1. Meilensteine für das neue System

Ich fertigte vor der ersten Analyse einen Katalog an, den ich mit Hilfe von am Markt befindlichen Lösungen erstellte. Ich ließ hier alle möglichen Gesichtspunkte einfließen, die ich auf der fertig gestellten Homepage wieder finden wollte. Die 3 Aspekte die hier behandelt wurden waren:

1. Der visuelle Aspekt der Startseite
2. Die Darstellung von News innerhalb der Startseite
3. Die Einbindung von Menüs für die weitere Navigation

Ad 1. Die visuellen Aspekte die wir uns für unsere Seite erarbeiteten waren in erster Linie die, dass man das Produkt welches man einmal gesehen hatte auch visuell wieder finden können sollte. D.h. für die Einbindung in der Startseite eine Anzeige zu generieren, wo alle Modelle im Mouse Over Effekt als freigestellte Grafik angezeigt werden und so einen Wiedererkennungseffekt auf der Homepage erzielen.

Ad 2. Der zweite Punkt die News mit einzubinden, sollte den lebendigen Effekt der Homepage erzielen. Man soll den Konsumenten über Neuerungen und Aktivitäten informieren und so eine Kontinuität der Produkte mittels der Startseite vermitteln.

Ad 3. Der letzte und auch schwierigste Punkt war die dezente und doch klare Gestaltung der Menüführung, nämlich einen goldenen Mittelweg zu finden aus welchen Navigationspunkten implementiert werden und auf welche verzichtet wird.

1.1.2.2. Analyse anderer Online Plattformen

Ein wesentlicher Bestandteil der ersten Analyse war es Internet Auftritte von anderen Marken und Herstellern zu analysieren. Wir lernten aus diesen Analysen einerseits eingesetzte Technologien kennen (soweit nachvollziehbar), ich lernte Stärken und Schwächen von User Interfaces kennen.

Erfahrungen aus den Analysen anonymisiert dargestellt:
Produkt 1: hier fand ich einen sehr guten Servicebereich, der durch klare Definition der Funktionalitäten herausragend war

Produkt 2: hier fand ich ein ausgezeichnetes Tool zur Berechnung der jeweiligen Preise eines Modells. Es war auch die Zusammenstellung von Grundmodell + dazugehörige Sonderausstattungen sehr komfortabel gestaltet.

Produkt 3: lebte lediglich von seiner sehr aufwendig gestalteten Produktdarstellung. Hier konnte ich einen sehr klaren Überhang in dieser Richtung feststellen welcher mir für meine Planungen klare Schranken aufzeigte.

Produkt 4: eine sehr gute interaktive Produktdarstellung die auch einen philosophischen Ansatz mit in der Implementierung zeigte, für meine späteren Vorstellungen eine nicht ganz optimale Lösung.

Produkt 5: hier fand ich ein internationales Portal vor, welches durch eine gute und klare Struktur realisiert wurde. Die Länderspezifische Trennung wurde sehr gut vollzogen. Die homogene Struktur der Darstellung der Daten wurde hier sehr einfach aber doch effizient gelöst.

Produkt 6: auf diesem Portal fand ich einen Auftritt mit starkem emotionalen Ausdruck. Die Kommunikation der Marke wurde hier sehr hervorgehoben. Der Anteil an Informationen für den Endkunden war hier sehr gering gehalten.

Aus den Analysen der 6 Produkte ergaben sich für mich folgende Punkte:

- Der durchschnittliche Standard der Webauftritte ist in diesem Bereich mittlerweile sehr hoch.
- Der Grossteil der Anbieter legt gesteigerten Wert auf umfassende Informations-Vermittlung
- Viele Plattformen in diesem Bereich bieten umfangreiche Servicefunktionalitäten
- Die Kommunikation der Markenstrategie und die emotionale Nutzerbindung haben branchenweit eine sehr hohe Bedeutung
- Die Herausforderung für den Konzern besteht darin, aufbauend auf den strukturellen und designseitigen Vorgaben diesen Standard zu erreichen und zu übertreffen

1.1.2.3.Ziele für das neue System

Durch die nachfolgende Grafik kann man die Zielvorgaben der neuen Web Plattform und deren primären Nutzen zum Ausdruck bringen:

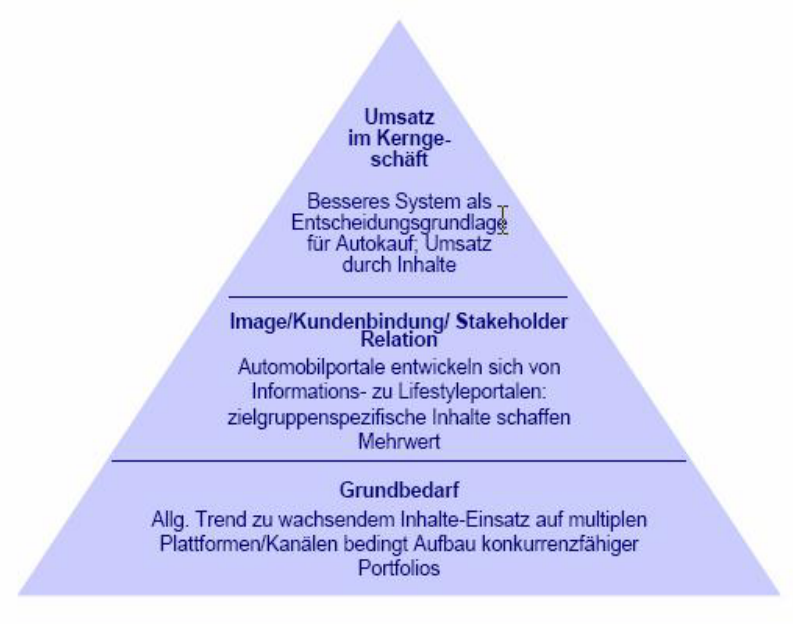


Abbildung 1.1: von mir definierte Bedürfnisse des CMS

Die oberste Schicht der Pyramide musste das Ziel sein, dem Kerngeschäft jedes Autohändlers nachzukommen, Autos zu verkaufen, darum sollte das System dem Kunden eine gute Entscheidungsgrundlage für den Kauf eines neuen Autos bieten. Die mittlere Schicht zeigt, dass eine Kundenbindung und eine Image Verbesserung durch das neue CMS System bewirkt werden sollen. Neue Komponenten wie die Darstellung von News, das Abbonement von Newslettern sollen das CMS System auch als Informationsportal stärken und den Kunden fühlen lassen, hier tut sich was. Der Grundbedarf als unterster Baustein ist einfach die Anforderung an den täglich steigenden Bedarf im Internet Informationen abzulegen. Diese Ablage der Informationen soll aber nach meiner Meinung in einer praktikablen Art und Weise geschehen und auch für normale User einfach zu bewerkstelligen sein.

Die Zielgruppen und deren Nutzen für dieses Portal sind klar festgelegt durch:

- Kunden
- Lieferanten

- Ausgewählte Zielgruppen

Abbildung 1.2: Zielgruppen und Nutzen des CMS

Die Grafik zeigt das Medium Web, welches durch Kunden, Lieferanten oder User besucht wird, die dann die jeweiligen Informationen abrufen.

1.2. Verstehen des Systemkontextes

Hier traf ich eine Auswahl von am Markt befindlichen CMS Systemen, die meinen Anforderungen unter funktionellen Aspekten entsprachen. Ich bewertete dann die Benutzerfreundlichkeit, die Möglichkeiten Updates einzupflegen aber auch die generelle Systemstabilität. Es sollte ausschließlich ein Freeware Produkt eingesetzt werden..

1.2.1. CMS Systeme Vorauswahl

Zu diesem Zeitpunkt waren für mich 3 Systeme in die engere Entscheidungsauswahl genommen worden.
Ich bewertete das System Typo3, das System phpCMS und das System von Zope.

Typo3 und phpCMS waren beide auf einer Php Basis aufgebaute Systeme, welches von den Erweiterungen her eine große Flexibilität versprachen.
Der Code der beiden Systeme war jedoch wie Tag und Nacht zueinander. Typo3 bot eine Vielzahl von Add On's im Auslieferungsumfang, das phpCMS war ein wenig schlanker und von den Funktionen nicht so umfangreich aufgestellt. Die beiden Hersteller siedelten beide Ihre Produkte für kleine Webseiten als auch für komplexe Firmenauftritte an.

phpCMS ist eine Mischung aus Template-Engine, Content Management System und Application Framework. Es erleichtert die Wartung komplexer Websites durch das automatische Aktualisieren der Sitemap, eine zentrale Verwaltung der Menüeinträge oder das Bereitstellen einer Volltextsuche. Änderungen am Layout oder der Menüstruktur müssen nur zentral an einer Stelle vorgenommen werden und wirken sich sofort auf alle Webseiten aus. Auch die Integration externer Anwendungen in

das Layout Ihrer Website ist dank direkter Einbindungsmöglichkeit für php-Scripte sowie dem Webgrab Modul, mit dessen Hilfe sie fast jede über HTTP erreichbare Applikation einbinden können (und das unabhängig von der verwendeten Programmiersprache) möglich. Viele PlugIns und Scripte erweitern phpCMS um nützliche Funktionen und Features und reduzieren so den Wartungsaufwand.

Typo3 besteht im Backend aus 3 Hauptbereichen: Dem Modul-Menü links, dem Seitenbaum in der Mitte und der Arbeitsfläche rechts. In der Web Ansicht erledigen Redakteure den überwiegenden Teil ihrer Arbeit. In der Seitenansicht können Seiteninhalte und -struktur editiert werden. Ganz intuitiv erscheint die Sitestruktur als "Baum", bei dem übersichtshalber nur die für den jeweiligen User editierbaren Seiten angezeigt werden. Setup Einstellungen, Zugangskontrolle uvm runden das reiche Angebot an Tools, Backends und Frontends dieses Systems ab.

Beide CMS Systeme haben meiner Meinung nach nicht die Sicherheit geboten die Notwendig ist, sie boten auch zu wenig Möglichkeiten das Layout betreffend und ließen mich zu dem Schluss kommen das zuviel Programmieraufwand nötig sein wird um die gewünschten Konzernvorgaben umzusetzen. Die Entscheidung viel auf Zope.

1.2.2. Zope Content Management System

Meine Wahl fiel auf das Zope Content - Management System. [Zope25]
Zope ist ein Rahmenwerk zum Erstellen von Web-Anwendungen. Eine Web-Anwendung ist ein Computerprogramm, auf das Benutzer mit einem Web-Browser über das Internet zugreifen. Sie können sich eine Web-Anwendung auch als eine dynamische Web-Site vorstellen, die den Benutzern nicht nur statische Information liefert, sondern Sie dynamische Werkzeuge verwenden lässt, um mit einer Anwendung zu arbeiten.

Web-Anwendungen gibt es überall und Web-Benutzer arbeiten ständig mit ihnen. Geläufige Beispiele für Web-Anwendungen sind Sites, mit denen Sie das Netz durchsuchen können, wie z.B. die Suchmaschine *Yahoo*, bei Projekten

zusammenarbeiten können, wie die Source Download Plattform *SourceForge*, über E-Mail mit anderen Leuten kommunizieren können, wie der Mail Anbieter *HotMail*. All diese Arten von Anwendungen können mit Zope erstellt werden. Die Vielfalt der Möglichkeiten und vor allem die bereits implementierten Systeme haben überzeugt.

Zope besteht aus mehreren verschiedenen Komponenten, die zusammenarbeiten, um zu helfen, Web-Anwendungen aufzubauen.

1.2.2.1.Komponenten von Zope

Zope kommt mit:

einem Web-Server

Zope enthält einen integrierten Web-Server, der den Benutzern Inhalt liefert. Natürlich kann man, wenn man diesen nicht benutzen will, einen schon vorhandenen Web-Server nutzen, wie in meinem Fall *Apache* oder auch *Microsoft IIS*. Man muss sich keine Sorgen machen. Zope arbeitet auch mit diesen Web-Servern und jedem anderen Web-Server der das Common Gateway Interface (CGI-Standard) unterstützt.

einer Web-basierten Benutzerschnittstelle

Wenn Sie Web-Anwendungen mit Zope aufbauen, verwenden Sie Ihren Web-Browser, um mit Zopes Verwaltungsschnittstelle, dem *Management Interface*, zu interagieren. Diese Schnittstelle ist eine Entwicklungsumgebung, die es erlaubt Dinge zu tun wie das Erstellen von Web-Seiten, das Hinzufügen von Abbildungen und Dokumenten, das Anbinden externer relationaler Datenbanken und das Schreiben von Skripten in verschiedenen Sprachen.

einer Objektdatenbank

Wenn Sie mit Zope arbeiten, arbeiten Sie hauptsächlich mit Objekten, die in Zopes Objektdatenbank gespeichert sind. Zopes Verwaltungsschnittstelle

liefert eine einfache, vertraute Art, Objekte zu verwalten, die etwa dem vorgehen bei üblichen Dateimanagern funktioniert.

relationale Integration

Sie müssen Ihre Informationen nicht in Zopes Objektdatenbank speichern, wenn Sie nicht wollen, weil Zope mit anderen relationalen Datenbanken wie *Oracle*, *PostgreSQL*, *Sybase*, oder wie in meinem Fall *MySql* und vielen anderen arbeitet.

Unterstützung von Scripting-Sprachen

Zope erlaubt Ihnen, Web-Anwendungen in einer Vielzahl von verschiedenen Sprachen wie Python, Perl, oder Zopes eigener "Document Template Markup Language" (DTML) zu schreiben.

Dies sind nur einige der Merkmale dieses Systems, welche Zope bei der Entwicklung von Web-Anwendungen so beliebt gemacht haben. Vielleicht ist Zopes bestes Feature aber seine Open-Source-Lizenz. Das heißt, nicht nur dass das Herunterladen von Zope kostenlos ist, sondern auch, dass Sie Zope ohne Lizenz Zahlungen oder Benutzungsgebühren in Ihren eigenen Produkten und Anwendungen verwenden dürfen. Zopes Open-Source-Lizenz bedeutet auch, dass der gesamte "Quellcode" von Zope für Sie offen liegt. Sie können Ihn überprüfen und erweitern. Zope zwängt Sie nicht in eine proprietäre Lösung, die Sie und Ihre Netzbenutzer in Ihren Bedürfnissen einschränkt.

Von der geschäftlichen Perspektive aus gibt es drei Schlüsselansätze, weshalb die Entscheidung auf Zope fiel: leistungsfähige Zusammenarbeit mit allen Komponenten und vor allem auch mit Standardkomponenten (wie Apache und MySql), ein einfaches Content-Management- und Web-Komponentensystem für die Einpflege und Wartung der laufenden Umgebung.

Um in dieser Umgebung flexibler zu sein, wurde Zope so geplant, um Steuerung sicher an verschiedene Gruppen von Benutzern auf jeder Ebene in der Web-Site delegieren zu können. Steuerung sicher zu weiter geben, heißt, folgende Dinge zu berücksichtigen:

1. Informationen auf leicht verständliche Weise zu präsentieren. Die meisten Leute verstehen es eher auf Ordner zu klicken, als Datenbankbefehle einzugeben, sodass Zope eine Schnittstelle verwendet, die einem einfachen Dateimanager ähnelt, wie dem *Microsoft Windows Explorer* oder anderen beliebten Dateimanagern.
2. Es kann schwierig sein, Programme für die Befehlszeile zu verwenden und Leute sind im Allgemeinen vertrauter im Umgang mit einem Web-Browser, sodass Zope dafür ausgelegt wurde, fast ausschließlich durch einen Web-Browser bedient zu werden.
3. Umgebungen für die Zusammenarbeit erfordern Hilfsprogramme, die es den Benutzern erlauben, ihre Fehler wieder rückgängig zu machen und zu arbeiten, ohne sich dabei gegenseitig zu behindern. Aus diesem Grund besitzt Zope eine Undo-Funktion, eine Versionsverwaltung und andere Hilfsprogramme, die Mitarbeitern dabei helfen, sicher zusammen zu arbeiten.

Diese Merkmale machen Zope zu einer idealen Umgebung für das Programmieren und das Verfassen von Web-Inhalten durch Gruppen und Untergruppen von Benutzern.

Viele Web-Anwendungen werden traditionell in drei Ebenen aufgebaut. Daten und andere Informationen sind in Datenbanken gespeichert, die Programme, die das Verhalten der Anwendung bestimmen sind in Dateien an einem Ort und HTML- und andere Layout- und Darstellungsinformationen an einem anderen gespeichert.

Dieses Vorgehen besitzt viele Vorteile, aber es hat auch Nachteile. Verschiedene Arten von Hilfsprogrammen und Grade des Sachverstands müssen verwendet werden, um mit den verschiedenen Komponenten zu arbeiten. Es kann sein, dass all die verschiedenen Komponenten ihre eigenen Überlegungen zu Sicherheit und Wartung bedürfen. Viele dieser Arten von Hilfsprogrammen sind von einem Web-Browser, oder von einfachen Befehlszeilen- oder GUI-Hilfsprogrammen wie FTP, aus nicht handhabbar.

In Zope sind all diese Komponenten zusammen in ein schlüssiges System gebracht. Alle erfordern einen gemeinsamen Satz an Diensten: Sicherheit, Web-Basierte

Verwaltung, Suche, Gruppierung, Verteilung und andere. Dadurch, dass es all diese Begriffe in ein handhabbares System bringt, ermöglicht Zope Ihnen, einen Satz an Fähigkeiten und einen an Werkzeugen zu verwenden, um komplexe Web-Anwendungen zu entwickeln. Außerdem bedeutet, mein Modell zu zentralisieren, dass Zope leichter mit anderen externen Hilfsprogrammen wie relationalen Datenbanken, GUI-Web-Editoren und anderen Systemen arbeiten kann, die mit Zope zusammenarbeiten.

Das Netz ist eine wachsende dynamische Plattform. Das Netz hat genug Standards und Programmierschnittstellen hervorgebracht, die es den Produzenten von Diensten, Produkten und Technologien erlauben, sich das Netz als ein architektonisches Modell vorzustellen, um das Sie ihre Anwendungen entwickeln können, anstelle bloß statische HTML-Dokumente an Benutzer weiterzugeben (mein Ausgangspunkt vor dem Projektbeginn).

Beweise dafür zeigen sich in vielen Standorten. Mit der ".*NET*"-Architektur stellt sich Microsoft eine Welt aus Netzkomponenten vor, die auf fernen Systemen laufen, während sie bestimmte Dienste an Anwendungen rund um die Welt liefern. *Frontier* von UserLand Software leistete Pionierarbeit für ein einfaches Protokoll für Web-Dienste, das XML-RPC genannt wird und Netzkomponenten erlaubt, miteinander zu kommunizieren (auch Zope arbeitet mit XML-RPC). Mit Web-Komponenten ist das Modell einer Person die vor dem Browser sitzt, nicht mehr das einzige Modell des Webs.

1.2.2.2. Entstehung von Zope

Die Entstehung von Zope: 1996 wurde Jim Fulton, Technikvorstand der Zope Corporation und Python-Guru, beauftragt, einer Schulklasse etwas über CGI-Programmierung beizubringen, obwohl er nicht viel über das Thema wusste. Jim Fulton studierte die gesamte vorhandene Dokumentation über CGI auf seinem Weg zum Unterricht. Auf dem Rückweg dachte Jim Fulton noch mal über alles nach was ihm an den traditionellen CGI-basierten Programmierumgebungen nicht gefiel: ihre Verwundbarkeit, ihr Mangel an Objektorientierung, und wie sie Details über den Web-Server nach außen hin sichtbar machen. Ausgehend von diesen

Anfangsüberlegungen wurde während des Rückflugs vom Unterricht im Flugzeug der Zope-Kern geschrieben.

Die Zope Corporation veröffentlichte weitere drei Open-Source-Softwarepakete, um das Web-Publishing zu unterstützen, *Bobo*, *Document Template* und *BoboPOS*. Diese Pakete waren in Python geschrieben. Sie haben sich zu Kernkomponenten von Zope entwickelt, die nun den ORB (Object Request Broker), die DTML-Scripting-Sprache und die Objektdatenbank liefern. Zope ist immer noch vorwiegend in Python geschrieben bis auf einige für die Leistung kritische Abschnitte in C. Python ist eine dynamisch objektorientierte Programmiersprache, die für viele Arten von Softwareentwicklung eingesetzt wird. Sie bietet sehr guten Komfort in der Anbindung an andere Sprachen und Tools, wird mit vielen Standard Libraries geliefert und ist einfach zu handhaben. Sie zeichnet sich auch durch effizienten Einsatz und in hohem Maß durch Zeitersparnis in der Entwicklung aus.

Damals hatte die Zope Corporation einen kommerziellen Anwendungs-Server basierend auf ihren drei Open-Source-Komponenten entwickelt. Dieses Produkt wurde *Principia* getauft. Im November 1998 überzeugte der Investor Hadar Pedahazur die Zope Corporation Principia zu Open-Source zu machen. Daraus wurde Zope, das sein eigenes Zuhause unter Zope.org bekam.

Man braucht viele Leute, die zusammenarbeiten, um eine Web-Anwendung zu erstellen. Diese Leute zu verwalten und zu koordinieren kann auf großen Sites eine schwierige Aufgabe sein. Ich habe einige gemeinsame Aufgabenbereiche in diesem Szenario identifiziert auf die man achten sollte:

- *Konsumenten* verwenden die Site, um nach nützlichem Inhalt zu suchen und damit zu arbeiten.
- *Geschäftliche Nutzer* erstellen und verwalten den Inhalt der Site.
- *Site-Designer* erstellen das Look-And-Feel der Site.
- *Site-Entwickler* programmieren die Dienste der Site.
- *Komponentenentwickler* erstellen Software für die spätere Weitergabe.
- *Administratoren* halten die Software und die Umgebung am Laufen.

- *Informationsarchitekten* treffen Plattformentscheidungen und behalten das Gesamtbild im Auge.

Zope ist eine Plattform, auf der Site-Entwickler Anwendungen erstellen, die an Site-Designer und Geschäftliche Nutzer weitergegeben werden und Komponentenentwickler geben neue Produkte und Anwendungen an Zope-Benutzer weltweit weiter.

Zope kann Zope-Produkte installieren, die sich auf unterschiedliche Zielgruppen konzentrieren. Zum Beispiel ist *Squishdot* ein beliebtes mit Zope verfasstes Weblog, das von Anfang an benutzt werden kann. Squishdot-Benutzer sehen nicht unbedingt, dass Zope unter der Haube steckt. Andere Zope-Produkte wie das "Content Management Framework" der Zope Corporation benutzen denselben Ansatz, indem sie auf Klienteln abzielen, die nichts von Zopes Existenz darunter wissen müssen.

1.2.2.3. Warum Zope

Wie können wir von Zope profitieren?

Ich habe mir Zopes Philosophie und seine Architektur angesehen. Lassen Sie uns jetzt einige von Zopes Einsatzmöglichkeiten untersuchen. Alle Sites lösen verschiedene Problemstellungen, aber viele Sites gehen täglich eine Vielzahl von Routine -Angelegenheiten an. Hier sind einige der Hauptverwendungen von Zope:

Dynamischen Inhalt darstellen

> Sie wollen die Präsentation Ihrer Web-Site für seine Benutzer Maßschneidern, Information in Datenbanken integrieren und Benutzern eine Suchfunktion bieten. Sie würden auch gerne Ihre Web-Site automatisieren und Ihre Geschäftsprozesse erleichtern. Kann Ihre Web-Site intelligent auf Besucher reagieren, um ihnen einen ansprechenden Eindruck zu vermitteln? Zope ermöglicht Ihnen, jede Art von dynamischen Seiten zu erzeugen. Es bringt bereits Funktionen zur Personalisierung, Datenbankintegration und Suche mit.

Ihre Web-Site verwalten

Eine kleine Web-Site ist leicht zu verwalten, aber eine Web-Site, die Tausende von Dokumenten, Abbildungen und Dateien bereitstellt, muss leistungsfähige Verwaltungswerkzeuge liefern. Können Sie die Daten, Anwendungslogik und Präsentation ihrer Site von einem Punkt aus verwalten? Können Sie mit ihren Inhalten Schritt halten oder gleiten sie Ihnen aus der Hand? Zope gibt Ihnen einfache und leistungsfähige Werkzeuge für das Bearbeiten von Gigabytes an Web-Inhalten. Sie können Ihre Logik, Ihre Darstellung und Ihre Daten komplett von Ihrem Web-Browser aus verwalten.

Ihre Web-Site sicher machen

Wenn Sie mit mehr als einer Handvoll Web-Benutzern umgehen, wird Sicherheit sehr wichtig. Es ist entscheidend, Benutzer zu verwalten, Aufgaben sicher an sie delegieren zu können. Zum Beispiel kann es sein, dass Mitarbeiter Ihrer technischen Abteilung in der Lage sein müssen, ihre Web-Seiten und Anwendungslogik selbst zu verwalten. Es kann sein, dass Entwickler Seitenvorlagen aktualisieren müssen und Datenbank-Administratoren Datenbankabfragen verwalten müssen. Kann Ihr System Tausende von Benutzern verwalten, eventuell mit Ihrem vorhandenen LDAP oder Ihren anderen Datenbanken über flexible Sicherheitsregeln verbunden? Zope erlaubt Ihnen, den Ihre Site für Tausende von Site-Managern und für Millionen von Besuchern anzupassen. Sie können Sicherheitseinstellungen einfach kontrollieren und sicher Steuerung an andere übertragen.

Netzwerkdienste anbieten

Die meisten Web-Sites bedienen jetzt Benutzer, aber Web-Sites müssen bald entfernte Computerprogramme und andere Web-Sites bedienen. Zum Beispiel würden Sie Ihre Nachrichten gerne automatisch für Web-Sites von Nachrichtenagenturen verfügbar machen. Oder vielleicht wollen Sie Produkte automatisch zum Verkauf auf einer durchsuchbaren Produktvergleichssite anbieten. Können Sie Ihre vorhandene Daten- und Anwendungslogik nutzen, um Netzwerkdienste zu erstellen, oder müssen Sie alles von Grund auf neu entwickeln? Zopes integrierte Netzfähigkeit macht jede Zope-Site zu einem

Netzwerkdienst. Auf Ihre Anwendungslogik und Ihre Daten kann über das Netz, über HTTP und XML-RPC zugegriffen werden.

Verschiedenartige Inhalte einzubinden

Ihr Inhalt liegt überall verstreut, in relationalen Datenbanken, Dateien, Web-Sites, FTP-Archiven, XML. Können Sie Ihre Daten zu einer schlüssigen Anwendung zusammenführen? Unterstützt Ihr System Web-Standards, sodass Sie den Inhalt von bereits vorhandenen und neuen Systemen, die Sie zukünftig hinzufügen integrieren können? Zope unterstützt Web-Standards, die Ihnen erlauben, Ihre vorhandenen Daten, Ihre vorhandene Infrastruktur und Ihre vorhandenen Dateisysteme zu verwenden.

Skalierbarkeit bieten

Sie haben also einen Glückstreffer gelandet und bekommen nun mehr Suchmaschinentreffer als Sie sich jemals vorgestellt hatten. Jetzt müssen Sie ein drastisch höheres Verkehrsaufkommen verarbeiten als zuvor. Können Sie Ihre Site auf eine andere Datenbank- und Serverplattform umstellen und die Belastung auf mehrere Server verteilen? Kann Ihre Web-Site wachsen, um Ihren Erfolg zu verkraften? Zope erlaubt Ihren Web-Anwendungen, über so viele Rechner verteilt zu arbeiten wie notwendig sind, dem Ansturm standzuhalten. Zope macht es möglich, eine kleine Site zu verwalten, die sich über Nacht in eine riesige Site verwandeln kann basierend auf seiner "ZEO"-Technologie.

Werfen wir einen näheren Blick auf die Zope Merkmale die uns erlauben, dynamische Web-Sites aufzubauen und zu verwalten.

Einzigartige Verwaltungsumgebung

Der Vorteil den wir in den ersten Schritten der Implementierung feststellten war, das Zope eine einfache Logik in seiner Darstellung der Daten hatte, und diese mit einem Web Browser einfach zu bedienen war. Das bedeutet weiters, dass Zope leicht zu verwenden ist und von großer Entfernung verwaltbar ist. Zope lässt Sie mit anderen zusammenarbeiten um interaktiv Ihre Web-Site zu entwickeln. (das war auch mit eine Entscheidungsgrundlage, das externe

Dienstleister und internes Personal die Pflege des CMS Systems übernehmen konnten und wir so ein hohes Maß an Flexibilität erreichten)

Integrierte Werkzeuge

Zope bietet Werkzeuge zur Site-Verwaltung, einen Web-Server, eine Suchmaschine, Datenbankverbindungen, Dienste für Sicherheit und die Zusammenarbeit und vieles mehr. Standardmäßig gibt Zope Ihnen alles, was Sie brauchen, um eine leistungsfähige Web-Site aufzubauen.

Unterstützung offener Standards

Zope sticht dadurch hervor, dass es aufgrund seiner Unterstützung für offene Standards unterschiedlichste Daten zusammenbringt. Zope unterstützt Internet-Standards einschließlich SQL, ODBC, XML, DOM, FTP, HTTP, FastCGI, XML-RPC, SOAP und andere.

Open-Source-Lizenz

Mit Zope bekommen Sie nicht nur eine Anwendung, Sie bekommen den Quelltext und eine Benutzergemeinde. Da Zope Open-Source ist, sind Sie nicht die Geisel einzelner Hersteller; Sie können Zope frei verwenden, verteilen und bearbeiten, um es Ihren Bedürfnissen anzupassen. Zope profitiert auch von einer aktiven Benutzer- und Entwicklergemeinde. Die Gemeinde verbessert Zopes Support, prüft Zopes Sicherheit, beseitigt Programmfehler und fügt neue Funktionen hinzu.

Erweiterbarkeit

Zope kann in viele Richtungen erweitert werden. Anwendungen von Drittherstellern können leicht erstellt und weitergegeben werden. Die Zope-Gemeinde hat Hunderte von Add-Ons für Zope produziert, von der Kreditkartenverarbeitung bis zu Web-Diskussionsforen.

Es sind viele Werkzeuge verfügbar, die Ihnen dabei helfen, Web-Anwendungen zu erstellen. Früh in der Geschichte des Webs wurden einfache Web-Anwendungen fast ausschließlich mit CGI-Programmen, die in Perl oder anderen Sprachen geschrieben

wurden aufgebaut. Jetzt gibt es eine Menge an Optionen, die sich von betriebsbereiten Open-Source-Rahmenwerken wie PHP, bis hin zu kommerziellen Optionen wie Cold Fusion, Java Application Servers und Vingettes Story Server erstrecken.

Zope bietet eine einzigartige Mischung von Merkmalen, einige etwas ähnlicher, und andere deutlich anders als die Merkmale, die andere Web-Anwendungs-Tools bieten. Zope ist leicht zu verwenden, Open-Source, leistungsfähig, und bietet Unterstützung für viele verschiedene Arten von Anwendungen. Hier ist eine kurze Liste von üblichen Nachteilen von Web-Hilfsprogramme und Zopes Vorteilen:

- Einige Hilfsprogramme bieten weder einen einfachen Dateimanager noch eine Benutzerschnittstelle und sind schwer zu verwenden. Zope hat eine einfache Benutzerschnittstelle.
- Einige Hilfsprogramme erfordern eine komplexe Konfiguration. Zope ist leicht zu installieren und erfordert keine Konfiguration, bevor Sie beginnen, es zu verwenden.
- Einige Hilfsprogramme erfordern die Verwendung ungewöhnlicher und proprietärer Entwicklungswerkzeuge. Zope arbeitet mit jedem Standard-Web-Browser und es werden keine anderen Hilfsprogramme benötigt, als sie auch zum Lesen dieses Buch verwenden.
- Einige Hilfsprogramme passen sich nicht wie Zope an, um eine große Anzahl von Entwicklern und Benutzern zu verwalten. Zope hat ein einheitliches, leistungsfähiges System zur Benutzerverwaltung, das sich bis hin zu vielen Benutzern mit eindeutigen, leicht verwaltbaren Berechtigungen anpassen kann.
- Zuletzt lassen sich kommerzielle Hilfsprogramme mit verschlossenem Quelltext nicht erweitern, anpassen oder weitergeben. Zope ist Open-Source.

Die Architektur von Zope grafisch dargestellt:

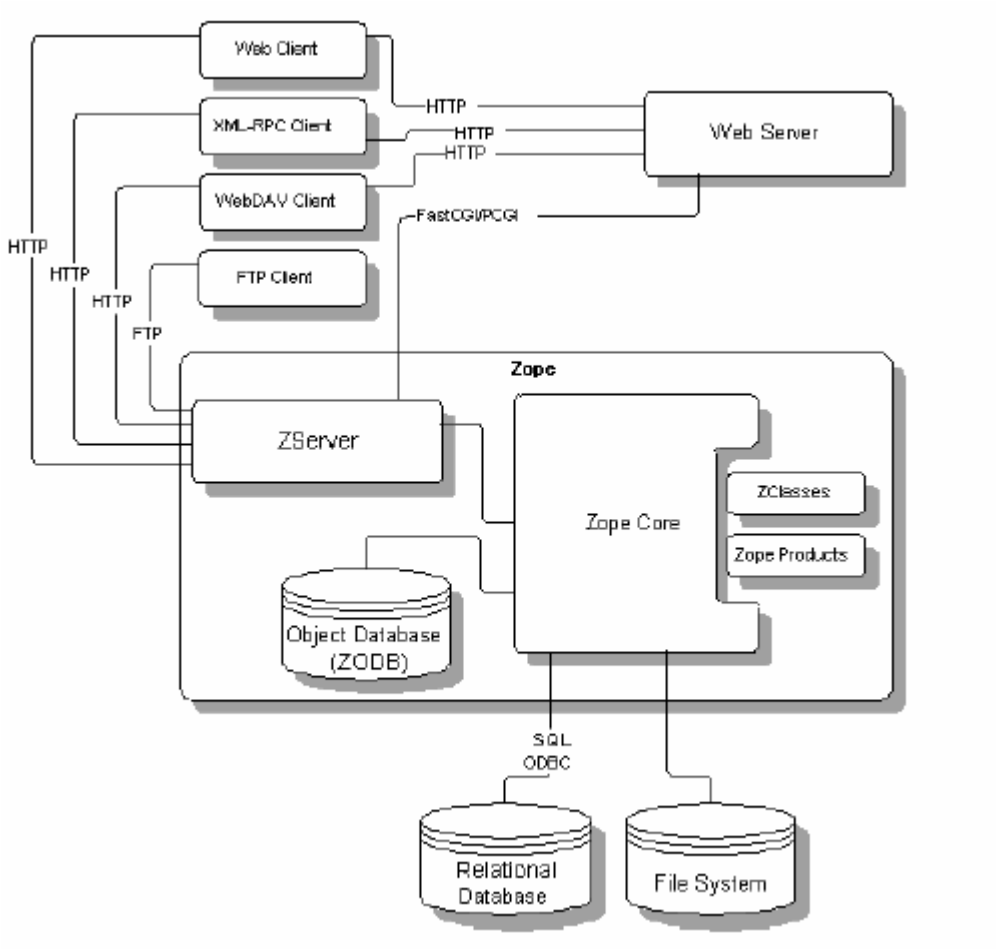


Abbildung 1.3: Architektur von Zope [Zope25]

Die einzelnen Komponenten der Grafik im Überblick:

ZServer — Zope kommt mit einem eingebauten Web Server, der den Content für Sie und Ihre User bereitstellt. Dieser Web Server dient weiters für FTP, WenDAV, und XML-RPC (eine remote procedure Call Funktion).

Web Server — Natürlich kann man auch die bestehenden Webserver einsetzten (wie *Apache* oder *Microsoft IIS)* und Sie können den mitgelieferten Web Server abdrehen. In meiner Lösung arbeiten wir mit beiden Servern parallel. Zope kann aber mit allen die das Common Gateway Interface unterstützen. (CGI).

Zope Core — Das ist die Engine die das Management Interface und die Objekt Datenbank koordiniert.

Object Database — Wenn man mit Zope arbeitet, werden normalerweise Objekte definiert, die in der Zope Objekt Datenbank gespeichert werden.

Relational database — Man muss die Objekte nicht in der von Zope zur Verfügung gestellten Zope Objekt Datenbank speichern, Zope unterstützt auch andere relationale Datenbanken wie *Oracle* , *PostgreSQL* , *Sybase* , *MySQL* und anderen.

File System — Zope kann natürlich auch mit Dateien arbeiten die in Ihrem Filesystem gespeichert sind und auf diese zugreifen.

ZClasses — Zope erlaubt Site Managern neue Objekttypen hinzuzufügen indem Sie das Zope Management Interface benutzen. ZClasses sind diese Artvon Objekten.

Products — Zope erlaubt Site Managern ebenso neue Objekttypen hinzuzufügen indem Sie neue “Product” files auf dem Zope File System Server installieren.

1.3. Ermittlung der funktionalen Anforderungen (welchen Server, Betriebssystem, Datenbanken)

Es wurde dann die Anforderung an die Hardware gestellt. Wie werden die Server (Main Server und Backup Server) konzipiert und in welchen Zeitraum.
Welches Betriebssystem wird ausgewählt (Suse Linux, SCO United Linux)
Auf dem Main Server habe ich mich für United Linux entschieden. United Linux wurde von den Marktführern Conective, der SCO Gruppe, SuSE und Turbolinux entwickelt. Diese 4 Unternehmen haben für das Projekt United Linux eine Kooperation gebildet um das Betriebssystem United Linux in höchster Qualität zu entwickeln. [UnLi1]

1.3.1. United Linux

United Linux definiert eine gemeinsame Basis (UL Basis) um der Gemeinschaft aller Linux Anbieter eine gemeinsame Basis für die Entwicklung zu liefern. Die Aufteilung des Core Systems vereinfacht den OEM’s und ISV’s die Zertifizierung

des Produktes und soll auch zu einer gemeinsamen Plattform führen und nicht zu einer Vielzahl von Distributionen.
Die teilnehmenden Partner konnten ihr „look&feel“ mit in die Entwicklung einbringen, hierfür war die Definition der UL-Basis das Herzstück der Entwicklung.
Wesentliche Vorteile von United Linux:

- Kombination von Linux Experten die an der Entwicklung teilnehmen bringt den Vorteil der gemeinsamen Knowledge Base die mit ins Produkt einfließt.
- Stabilität: Aufgebaut auf der soliden Basis der Foundation bietet United Linux ein Operating System mit großer Zuverlässigkeit und Stabilität.
- Qualitätssicherung: getestet von QA teams und Zertifizierungslabors weltweit, kann dieses Open Source operating system mit einem Qualitätslevel aufwarten, der sonst nur von teuren propriäteren Systemen erwartet werden darf.
- Zertifizierung: Sicherstellung der Zertifizierung durch alle namhaften Hard- und Softwarehersteller, es bietet eine hervorragende Umgebung für alle Anwendungen mit sich.
- Weltweiter Vertrieb: United Linux ist weltweit präsent und kann überall Supportet werden

Die Architektur von United Linux [UnLi1]:

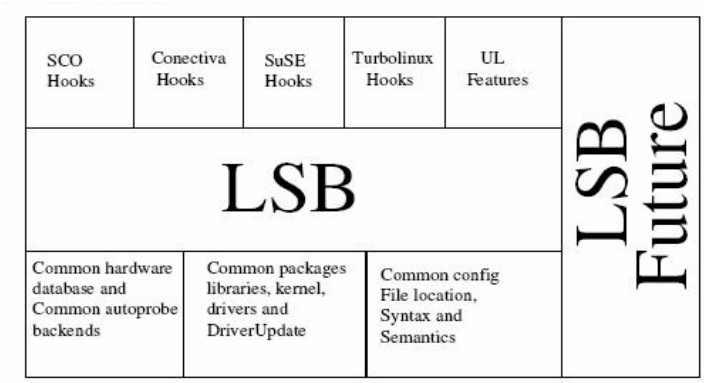


Abbildung 1.4: United Linux System Architektur, (LSB – Linux Standard Base)

United Linux bietet eine Vielzahl an Standard und Notfall Hard- und Softwaretechnologien die nachfolgend kurz definiert sind:

- Automatisierte Installation: der Installer kann XML Optionen lesen. Installationsmethoden sind con CD, NFS mounted Directory, local Harddisk Partition
- Hochverfügbarkeit, Journaling Filesystem, LVM
- NGPT Next Generation Possix Threads, bringt eine Verbesserung für Applikationen die mit pthreads arbeiten. Auch AIO Asynchronous I/O wird unterstützt
- Raw I/O and Direct I/O (direktes speichern von Buffer Daten ohne Kernel)
- Hyper Threading um die response Zeiten zu verbessern
- SNMP, Large Memory Support, IPv6, Directory Service (LDAP)

In United Linux vorhandene Serverprogramme, Services and Applications [UnLi2]

- Web server (Basis Pakete): Apache Web Server, Apache Extension Module, PHP, PHP Extensions, Tomcat Server
- File and Print: Windows (Samba), Mac (net talk), UNIX (LPR, NFS)
- Name Server and Internet/Intranet connection Server: DNS, WINS (Samba), DHCP, FTP, TFTP
- Mail and news Server: SMTP (postx, sendmail), POP, IMAP
- Proxy Server: Squid
- SQL Database Server
- Andere: NIS, GNOME, Kerberos, KDE, Mozilla

Auf dem Backup Server habe ich mich für SUSE Linux in der Version 9.1 entschieden. (heutige Version Suse Linux 10.0) [SuseLi1]
SUSE Linux wurde 1992 in den Markt eingeführt und seit diesem Zeitpunkt immer weiter entwickelt und wurde zu einer starken Alternative am Linux Markt.
Die wesentlichen Merkmale von SUSE Linux 9.1 sind:

- Der neue Linux Kernel 2.6 mit seiner verbesserten Performance, Skalierbarkeit und Speicherverwaltung für Prozesse. Ebenso kommen Weiterentwicklungen in der Video und Audio Integrität im Kernel hier erstmals voll zu tragen. Durch die gesteigerte System Performance sollen auch Audio und Video Streaming substantiell verbessert sein.

- Eine neue Samba Version die zu einer besseren Windows Connectivity beitragen soll. Ebenso stellen die Implementierung von Active Directoy und die Konfiguration von virtuellen Servern eine Verbesserung der Anbindung dar.
- Support von Windows NTFS Filesystem, welches von W2k und XP verwendet wird
- Neuer KDE (3.2.1) und GNOME (2.4.2)

Vieler Studien zu Grunde gelegt, wählte auch ich den Weg, eine vollständige so genannte LAMP (Linux – Apache – MySql – PHP) Umgebung auf den Servern einzurichten. Daher fiel auch die Entscheidung auf MySql als Datenbank.

1.3.2. Mysql

Der Datenbankserver enthält alle in der Geschäftswelt geforderten Funktionen und viele Neuheiten:

- Mehrere Speicher-Engines einschließlich vollständiger Transaktionsunterstützung mit Commit, Rollback, Absturz-Wiederherstellung und der Möglichkeit, Datensätze auf Zeilenebene zu sperren. Ihnen steht eine Auswahl an Speicher-Engines zur Verfügung. Sie können von schnellen Speicherbasierten Techniken über vollständige Transaktionskontrolle bis hin zu Cluster-Lösungen für Hochverfügbarkeit wählen.
- Der Abfrage-Cache stellt deutliche Leistungsvorteile zur Verfügung. In Verbindung mit Datenbank-Replikation können Sie durch die Nutzung mehrerer Slave-Server Geschwindigkeit und Robustheit Ihres Systems erhöhen.
- Ein stabiles Sicherheitssystem mit fortschrittlichen Zugriffsrechten und Unterstützung von SSL-Verschlüsselung stellt robuste Anwendungssicherheit zur Verfügung.
- Volltextindizierung und -suche ermöglichen in Textfeldern die schnelle Suche nach Wörtern oder Sätzen. Dabei wird Ihnen auch die Relevanz der Ergebnisse angezeigt. Auch genaue Satzvergleiche und Boolesche Operatoren sind möglich.

In dem Bericht "MySql Breaks Into the Data Center" hat die Computerworld dargelegt, wie MySql zur populärsten Open-Source-Datenbank der Welt wurde und warum Unternehmen beabsichtigen, ihre Betriebskosten durch die Verwendung von MySql zu senken. [Mysql1]

MySql reduziert die Datenbankbetriebskosten Ihrer Anwendung durch:

- Senkung der Datenbank-Lizenzkosten um über 90%
- Verringerung der Ausfallzeiten um 60%
- Minderung der Hardware-Aufwendungen um 70%
- Verringerung der Verwaltungs-, Technik- und Wartungskosten um bis zu 50%

3 Hauptgründe für die Entscheidung für die populärste Open Source Datenbank sind:

- MySql ist eine schnelle, einfach zu bedienende verlässliche Datenbank Entwicklung welche um einen Bruchteil der Kosten einer propriäteren Software als Freeware zu haben ist. Die Features entsprechen unseren gesetzten Anforderungen an Verfügbarkeit und Skalierbarkeit.
- MySql hat mehr als 6 Millionen Installationen weltweit und wird mehr als 30.000x täglich downgeloaded.
- MySql wird durch MySql AB supportet, einer Open source Firma die 1995 gegründet wurde

Die Kostenentscheidung lässt sich wie folgt darstellen [Mysql1]:

Abbildung 1.5: Kosten von DB Systemen

Die Antwortzeiten lagen im Vergleich wie in Abbildung 5 zu sehen ist:

Database Server	Response Time in Seconds
MySQL 4.0.1	31
IBM DB2 7.2 Fix Pack 5	102
MS SQL Server 2000 Ent. Ed. SP2	109

Abbildung 1.6: Antwortzeiten im Vergleich

Mysql Version 4.0.1 hatte bei dem Test eine Durchschnittlich Antwortzeit von 31 sec. Die Datenbank von IBM DB2 und MS SQL Server lagen beide über 100 Sekunden, welches auf eine gute Performance der Datenbank schließen lässt.

2. Analyse und Design:

2.1. Analyse, Verfeinerung und Erfassung unberücksichtiger Anforderungen

Es sollen hier die nicht transparenten Funktionalitäten angesprochen werden wie z.B. das ein Brand- Verantwortlicher direkt via User Interface das CMS System bedienen können soll. Es soll auch eher ohne externe Beteiligung das System am Leben erhalten werden. Hier soll unter Betracht der Kosten ein ausgewogenes System gefunden werden. (Kosten Nutzen Faktor einbinden) Ich wollte mit dem neuen CMS System auch erreichen, das die Autoren auch in der Lage waren eigene Artikel in das System einzupflegen. Unter eigenen Artikeln darf man hier aber nicht das Schreiben eines einfachen News Text verstehen sondern die Abbildung von Tabellen, Texten und Bildern auf den einzelnen Seiten.
Ich löste diese Anforderung einfach mit einem System von Vorlagen für die Autoren, die durch Stylesheets umgesetzt wurden.

Als erstes baute ich ein Stylesheet auf, in welchen die für den Content notwendigen Parameter vordefiniert wurden.
CSS Stylesheets sind eine unmittelbare Ergänzung zu HTML[Css1]. Es handelt sich dabei um eine Sprache zur Definition von Formateigenschaften einzelner HTML-Elemente. Mit Hilfe von Stylesheets können Sie beispielsweise bestimmen, dass Überschriften 1. Ordnung eine Schriftgröße von 18 Punkt haben, in der Schriftart Helvetica, aber nicht fett erscheinen, und mit einem Abstand von 1,75 Zentimeter

zum darauf folgenden Absatz versehen werden. Angaben dieser Art sind mit reinem HTML nicht möglich.
Das ist aber nur der Anfang. Stylesheets bieten noch viel mehr Möglichkeiten. So können Sie HTML-Elemente - egal ob Textabsätze, Listen, Tabellenzellen oder Formulare - mit einer eigenen Hintergrundfarbe, einem eigenen Hintergrundbild (Wallpaper) oder mit diversen Rahmen ausstatten. Sie können Elemente pixelgenau im Anzeigefenster des WWW-Browsers positionieren. Für Print-Layouts stehen Möglichkeiten zur Definition von Seitenlayout und Textflusskontrolle bereit. Für die akustische Wiedergabe von Web-Seiten gibt es ein ganzes Arsenal an Befehlen, um die künstliche Sprachausgabe fernzusteuern. Einige CSS-Eigenschaften sind auch in der Lage, das Anzeigefenster des Browsers zu beeinflussen, so etwa das Aussehen des Cursors. Spezielle Filter schließlich, die allerdings rein Microsoft-spezifisch sind, erlauben Grafik-Effekte bei normalen Texten, die aus Grafikprogrammen wie Photoshop bekannt sind.
Ein weiteres wichtiges Leistungsmerkmal von CSS ist die Möglichkeit, zentrale Formate zu definieren. So können Sie beispielsweise im Kopf einer HTML-Datei zentrale Definitionen zum Aussehen einer Tabellenzelle notieren. Alle Tabellen-zellen der entsprechenden HTML-Datei erhalten dann die Formateigenschaften, die einmal zentral definiert sind. Das spart Kodierarbeit und macht die HTML-Dateien kleiner. Sie können Ihre Stylesheet-Definitionen sogar in separaten Dateien notieren. Die Stylesheet-Dateien können Sie in beliebig vielen HTML-Dateien referenzieren. Auf diese Weise können Sie für große Projekte einheitliche Layouts entwerfen. Mit ein paar kleinen Änderungen in einer zentralen Stylesheet-Datei können Sie dann für hunderte von HTML-Dateien ein anderes Aussehen bewirken.
CSS Stylesheets unterstützen also erstens die professionelle Gestaltung beim Web-Design, und zweitens helfen sie beim Corporate Design für große Projekte oder für unternehmensspezifsche Layouts.
Im Anhang III unter Listung: 2.0: Stylesheet für Content Modul finden Sie das von mir gestaltete CSS Stylesheet für das CMS System im Detail. (hier nur wenige Zeilen)

```
body {margin:0px;padding:0px;
      background-color:#dadccf;
      background-image:url(images/bg_lines.gif)}
img, table {border:none;margin:0px;padding:0px}
```

```
p, td, .sm_subnav, .sm_subsubnav, .nav {font-
family:Arial;font-size:12px;font-weight:normal;text-
decoration:none;margin:0px;padding:0px}
.nav { font-size: 11px; color: white; }
/*#head */
#head { /*background:url(images/head.gif);background-
repeat:no-repeat;*/ height:82px;}

#logo {position:absolute; top:23px; left: 45px; }
```

Listung: 2.0: Stylesheet für Content Modul

Nach Fertigstellung der Layout Definitionen wurden die ersten Templates für die Darstellung von Content entworfen. Es wurden folgende Möglichkeiten entwickelt:

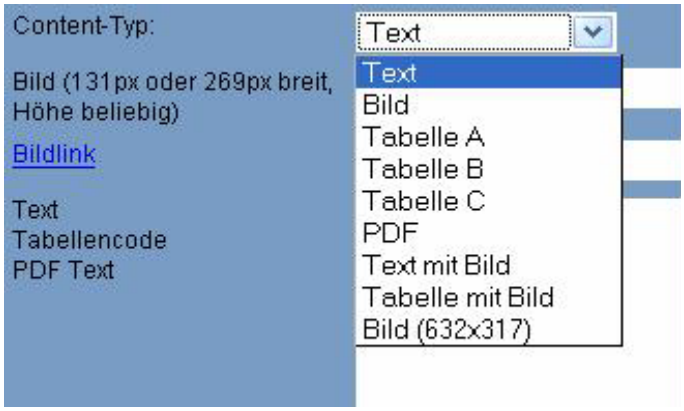


Abbildung 2.1: Layout Typen

Jede der möglichen Layouttypen (Text, Bild, Tabelle A, Tabelle B, Tabelle C, PDF, Text mit Bild, Tabelle mit Bild und Bild) hatte ein eigenes programmiertes Aussehen.

Der Code des Tabellentyp A sieht wie folgt aus:

```
<dtml-if tableTitle><p style="padding:8px 8px 0px
8px"><b><dtml-var tableTitle></b></p></dtml-if>
<div id="<dtml-if sequence-odd>ctbg-odd<dtml-var
getCSSClass><dtml-else>ctbg-even<dtml-var getCSSClass></dtml-
if>" style="margin:8px 0px 0px 0px;">
<table cellpadding=0 cellspacing=0 style="width:632px">
<dtml-in "_.string.split(content_txt, '\n')">
<dtml-if sequence-start>
<dtml-call "setTableAlignVar(_['sequence-item'])">
<dtml-elif "_['sequence-number'] == 2">
  <tr>
  <dtml-in "_.string.split(_['sequence-item'], ';')">
  <td class=thl <dtml-try>align=<dtml-var "align[_['sequence-
index']]"><dtml-except></dtml-try>><dtml-var
"parseTableContent(_['sequence-item'])"></td>
  <td class=thlspacer></td>
  </dtml-in>
```

```
  </tr>
<dtml-else>
  <tr>
  <dtml-in "_.string.split(_['sequence-item'], ';')">
    <td class=<dtml-if sequence-odd>todd<dtml-
else>teven</dtml-if> <dtml-try>align=<dtml-var
"align[_['sequence-index']]" ><dtml-except></dtml-try>>
      <dtml-var "parseTableContent(_['sequence-item'])"></td>
    <td class=lines></td>
  </dtml-in>
  </tr>
<dtml-if sequence-end><dtml-else>
  <tr><td class=spacer colspan=<dtml-var
"_.len(_.string.split(_['sequence-item'], ';'))*2-
1">></td></tr>
</dtml-if>
</dtml-if>
</dtml-in>
</table>
</div>
```

Listung 2.2: Definition Tabellentyp A

Hier wird z.B. festgelegt, dass die Tabelle einen Rahmen besitzt, die erste Zeile der Tabelle in dunkelgrau dargestellt wird und die Zeilen abwechselnd in Weis und Hellgrau dargestellt werden.

Der Autor gibt die dazugehörigen Daten wie folgt ein:

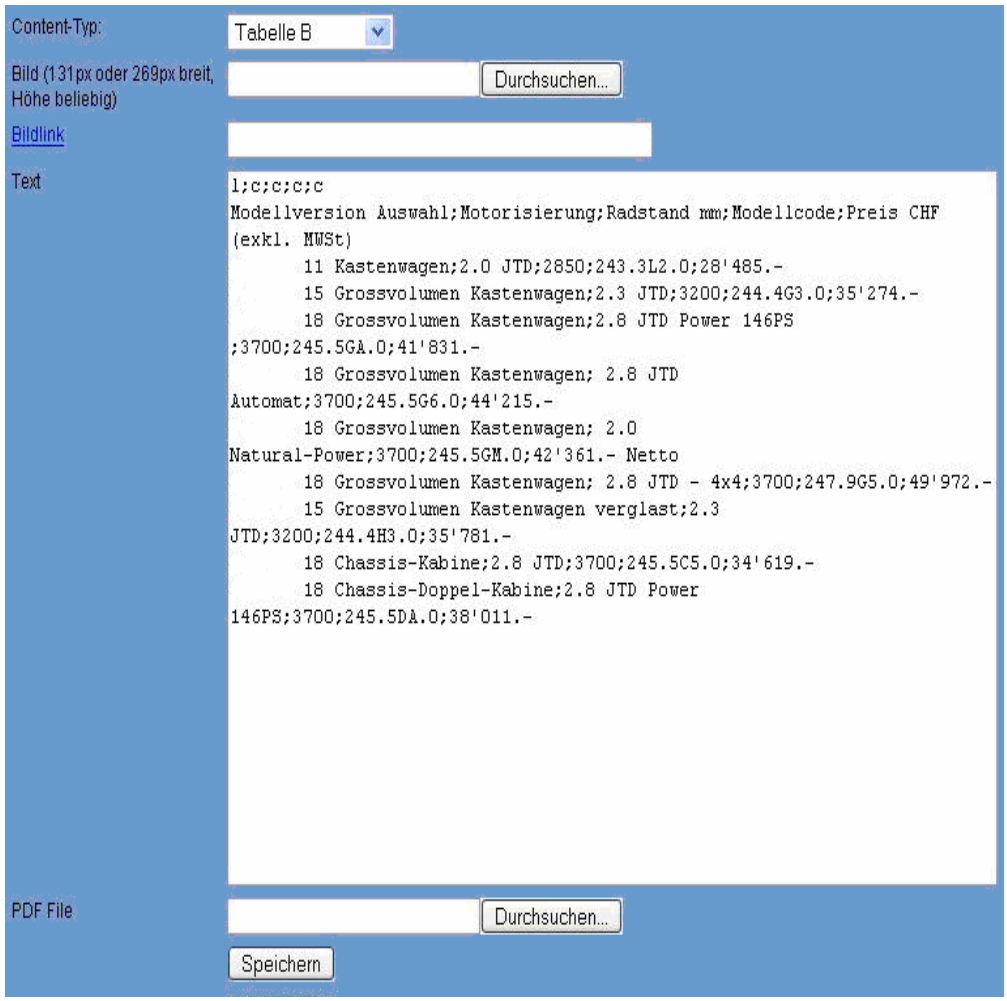


Abbildung 2.3: Grafik der Dateneingabe

Der Autor entscheidet sich zuerst für einen Content Typ. Hier ist es der Tabellentyp B. In weiterer Folge wird unter Text der dazugehörige Content erfasst.

In Zeile 1: l;c;c;c;c werden die Anzahl der Spalten und Ihre Ausrichtung definiert (links, Center), 5 Spalten

In Zeile 2: die Texte der Überschriften

Ab Zeile 3: die Inhalte der Spalten

Das Ergebnis wird dann auf der Webseite wie folgt dargestellt:

Modellversion Auswahl	Motorisierung	Radstand mm	Modellcode	Preis CHF (exkl. MWSt)
11 Kastenwagen	2.0 JTD	2850	243.3L2.0	28'485.-
15 Grossvolumen Kastenwagen	2.3 JTD	3200	244.4G3.0	35'274.-
18 Grossvolumen Kastenwagen	2.8 JTD Power 146PS	3700	245.5GA.0	41'831.-
18 Grossvolumen Kastenwagen	2.8 JTD Automat	3700	245.5G6.0	44'215.-
18 Grossvolumen Kastenwagen	2.0 Natural-Power	3700	245.5GM.0	42'361.- Netto
18 Grossvolumen Kastenwagen	2.8 JTD - 4x4	3700	247.9G5.0	49'972.-
15 Grossvolumen Kastenwagen verglast	2.3 JTD	3200	244.4H3.0	35'781.-
18 Chassis-Kabine	2.8 JTD	3700	245.5C5.0	34'619.-
18 Chassis-Doppel-Kabine	2.8 JTD Power 146PS	3700	245.5DA.0	38'011.-

Abbildung 2.4: Tabellentyp B fertig dargestellt

Ein weiteres Beispiel mit größeren Datenmengen aber dem gleichen Tabellentyp:

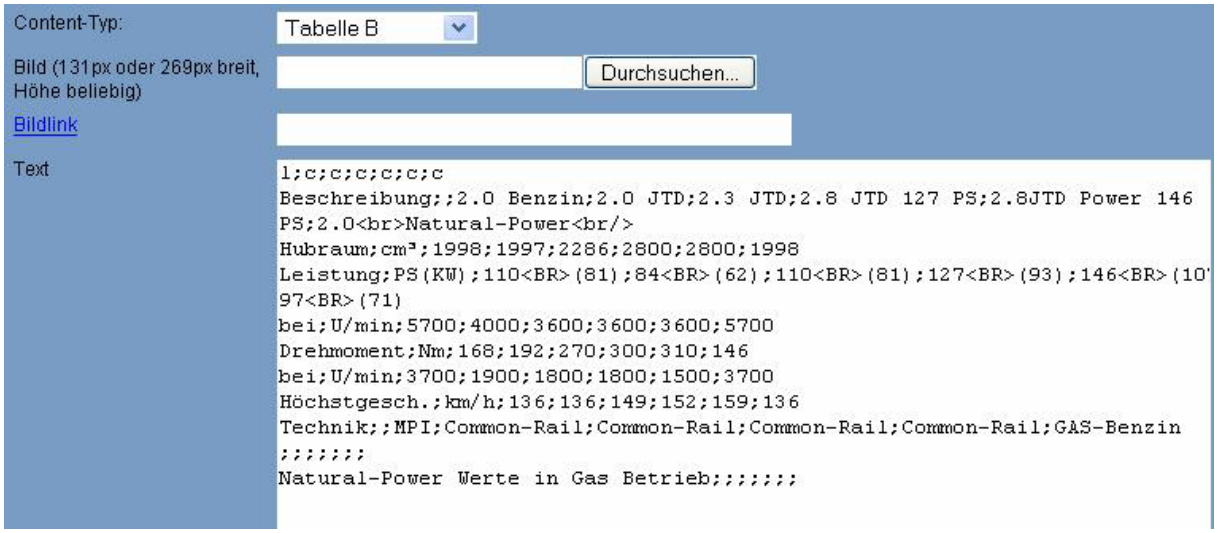


Abbildung 2.5: Tabellentyp B mit mehr Daten

Und das Ergebnis dann auf der Webseite

Beschreibung		2.0 Benzin	2.0 JTD	2.3 JTD	2.8 JTD 127 PS	2.8JTD Power 146 PS	2.0 Natural-Power
Hubraum	cm³	1998	1997	2286	2800	2800	1998
Leistung	PS(KW)	110 (81)	84 (62)	110 (81)	127 (93)	146 (107)	97 (71)
bei	U/min	5700	4000	3600	3600	3600	5700
Drehmoment	Nm	168	192	270	300	310	146
bei	U/min	3700	1900	1800	1800	1500	3700
Höchstgesch.	km/h	136	136	149	152	159	136
Technik		MPI	Common-Rail	Common-Rail	Common-Rail	Common-Rail	GAS-Benzin
Natural-Power Werte in Gas Betrieb							

Abbildung 2.6: Tabellentyp B fertig dargestellt

Zum Unterschied möchte ich noch den Content Typ Bild zeigen.

Der Code des Content Typs Bild sieht wie folgt aus:

```
<dtml-if content.gif>
      <dtml-if *link==''*>
            <img src="&dtml-absolute_url;/content.gif"
alt="Image">
      <dtml-else>
            <a href="&dtml-link;" target="<dtml-var
„ckeckTarget(link)">"><img src="&dtml-
absolute_url;/content.gif" alt="Image"></a>
      </dtml-if>
</dtml-if>
```

Listung 2.7: Content Typ Bild

Der Code des Content Typ Text sieht wie folgt aus.

```
<dtml-var „parseStructuredText(content_txt)" fmt=structured-
text>
```

Listung 2.8: Content Typ Text

Dieser Content Typ wurde auch herangezogen um z.B. eine Flash Animation auf die Webseite zu integrieren.

Der Autor musste dazu folgende Syntax eingeben:

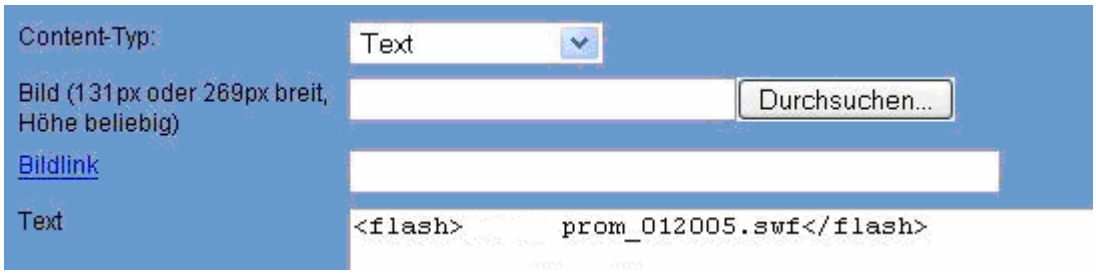


Abbildung 2.9: Content Typ Text mit Flash Inhalt

Auf der Webseite wurde dann dieses Flash File fertig umgesetzt angezeigt:

Abbildung 2.10: Text Content als Flash

Die grundsätzliche Navigation in den einzelnen Kategorien wurde vom Grundsystem selber zur Verfügung gestellt. Ich brauchte den User nur noch mit dem Aussehen und den Funktionalitäten vertraut zu machen um zu den Unterseiten zu gelangen wo der Content einzugeben war.

Die Editier Funktion sah wie folgt aus:

Abbildung 2.11: Editiersyntax

Wir sehen hier den Auszug aus der linken Navigationsleiste (leftnav). Der User erhält pro vorhandenen Eintrag im System kleine Buttons um diese zu modifizieren, zu löschen oder neu zu platzieren.
Die Bedeutung der Zeichen im einzelnen:
1. Symbol: diente zum Löschen des Eintrages
2. Symbol: diente zum online / offline stellen des Menüpunktes / Contents (grün ist online, rot ist offline)
3. + 4. Symbol: um die Reihung der Inhalte zu verändern
5. Symbol: um den Eintrag zu editieren
6. Symbol um einen Untermenüpunkt zu erstellen.

In dieser Weise und mit diesen geschaffenen Möglichkeiten wurde einerseits ein Gleichgewicht für den Autor (einen nicht ausgebildeten Informatiker) und den Informatikern der Entwicklung geschaffen, andererseits auch eine einfache aber doch sehr flexible Umgebung für die Gestaltung der Webseiten geschaffen die durch einfache Modifikation von CSS Stylesheets oder von Anlage neuer Content Typen auch schnell in ein neues Erscheinungsbild schlüpfen konnten.

3. Implementierung des Gesamtsystems:

3.1. Aufbau des Grundsystems

Stück für Stück werden die gestellten Anforderungen umgesetzt und ein Teilsystem nach dem anderen geht online und kann indirekt getestet werden.
Nach Installation des Grundsystems werden die ersten Hauptnavigationsleisten in das leere Content Management System eingefügt.
Im zweiten Schritt werden die Knotenseiten errichtet und Ihre Funktionalitäten festgelegt. Die Knotenseiten sind das Kernstück der Navigation und verzweigen zum Beispiel in die Modellauswahl, die Detailansicht der Produkte oder das wichtigste Instrument für das Marketing die Prospektbestellung.

3.2. Teilsystem des Marketing

Dieser Teilbereich stellt eine Kernaufgabe des Content Management Systems dar, der hier etwas genauer erklärt wird: den Teil der Kundenbindung und Adress-gewinnung via Webseite, die Prospektbestellung.

Die Prospektbestellung stellt eine Zusammenfassung aller auf der Seite dargestellten Modelle dar. Es musste nach jedem Produkt ausgewählt werden können. Die einzelnen Modelle konnten in Checkboxen angewählt werden.
Im 2. Teil der Seite wurden dann Anrede, Vorname, Nachname, Plz und Ort erfasst. Eine weitere Checkbox stand für die Abbonierung des Newsletters. Das Layout der Seite wurde wie folgt dargestellt:

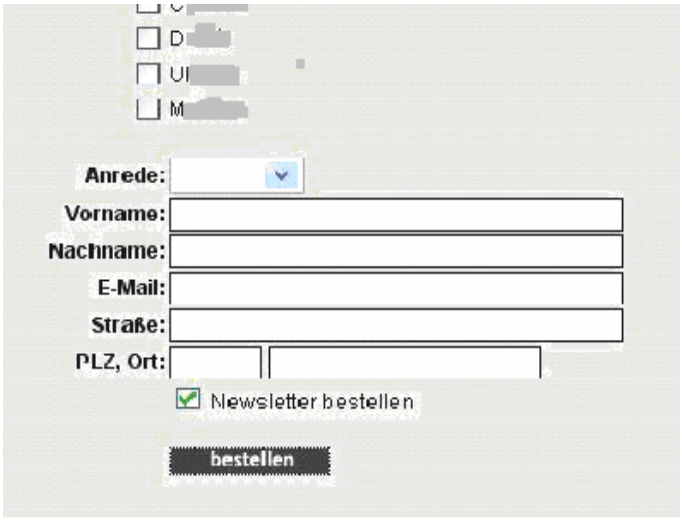


Abbildung 3.1: Layout Newsletter

Die eigentliche Arbeit der Webseitenprogrammierung war aber hinter dem Button Bestellen verborgen.
Nach dem klicken des Button Bestellen wurden folgende Aktionen durchgeführt:

- Eintrag in eine Adressdatenbank
- Versenden eines Emails an eine Agentur
- Versenden eines Emails an den Brandverantwortlichen
- Eintrag der Statistik Daten

Der Eintrag in die Adressdatenbank wurde einerseits für die Versendung des Newsletters genutzt, aber auch für postalische Versendungen. Diese Daten stellen natürlich den größten Wert für ein Unternehmen dar, in Punkto Kundenneuge-winnung, Interessentengewinnung und Datenanalysen auf die einzelnen Segmente der Kunden.

Das versenden der Daten an die Agentur und den Brandverantwortlichen diente einerseits dazu der Person das gewünschte Prospekt zuzustellen, andererseits die Daten intern zu nutzen und ein Feedback über das Interesse an einem Produkt zu haben.

Die statistischen Daten sollen Auskunft geben, wie viele Newsletter bestellt wurden, über welches Produkt Informationen angefordert wurden und über die generellen Zugriffe auf die Seite.

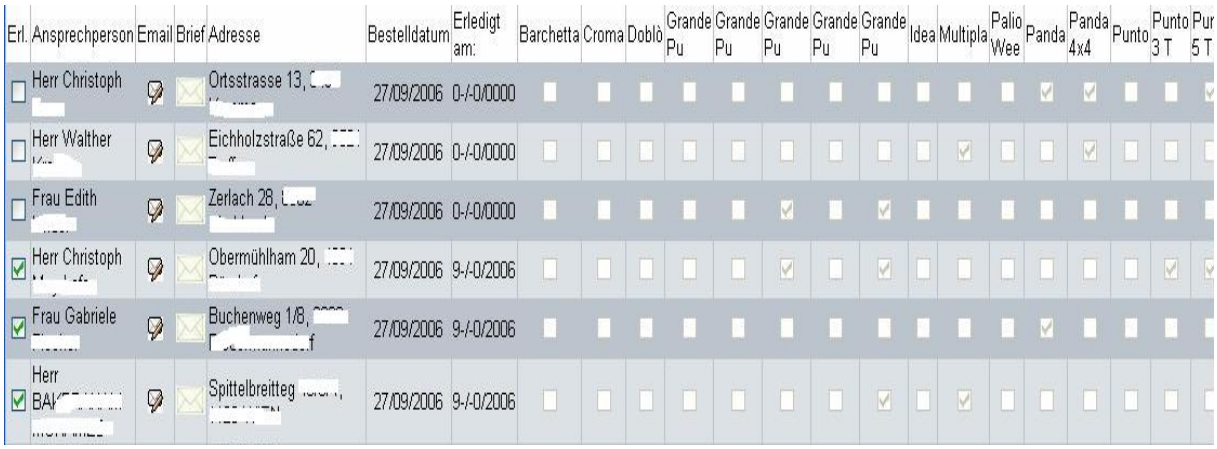
Der Code der Seite sieht wie folgt aus: Details des Codes sind im Anhang III , Listung 3.2: Code des Newsletter im Auszug, hier nur wenige Zeilen des Java Scriptes:

```
<script language="JavaScript" type="text/JavaScript">
<!--
function MM_findObj(n, d) { //v4.01
  var p,i,x;  if(!d) d=document;
if((p=n.indexOf("?"))>0&&parent.frames.length) {
    d=parent.frames[n.substring(p+1)].document;
n=n.substring(0,p);}
  if(!(x=d[n])&&d.all) x=d.all[n]; for
(i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
  for(i=0;!x&&d.layers&&i<d.layers.length;i++)
x=MM_findObj(n,d.layers[i].document);
  if(!x && d.getElementById) x=d.getElementById(n); return x;
}
```

Listung 3.2: Code des Newsletter im Auszug

Für die Marketing Abteilung wurde ein eigenes Tool entwickelt um den Usern ein geeignetes Instrument zur Kontrolle der Arbeit der Agentur zu geben und um eine Übersicht der Anfragen der Kunden zu haben.

Die Maske zur Kontrolle der Daten war wie nachfolgend abgebildet aufgebaut:



Erl.	Ansprechperson	Email	Brief	Adresse	Bestelldatum	Erledigt am:	Barchetta	Croma	Doblò	Grande Pu	Grande Pu	Grande Pu	Grande Pu	Grande Pu	Idea	Multipla	Palio Wee	Panda	Panda 4x4	Punto	Punto 3 T	Pur 5 T
☐	Herr Christoph			Ortsstrasse 13,	27/09/2006	0-/-0/0000																
☐	Herr Walther			Eichholzstraße 62,	27/09/2006	0-/-0/0000																
☐	Frau Edith			Zerlach 28,	27/09/2006	0-/-0/0000																
☑	Herr Christoph			Obermühlham 20,	27/09/2006	9-/-0/2006																
☑	Frau Gabriele			Buchenweg 1/8,	27/09/2006	9-/-0/2006																
☑	Herr BA			Spittelbreitteg	27/09/2006	9-/-0/2006																

Abbildung 3.3: Layout Datennachbearbeitung

Die Agentur trug das Versanddatum des Prospektes ein und markierte die erste Spalte mit einem Hakerl, dass diese Anfrage erledigt sei.
Die Marketingabteilung Ihrerseits konnte nun eigenständige Abfragen nach Modellen erstellen und auch andere Daten wie Adresse und email für Marketingzwecke weiter benützen.

3.3. Teilsystem des Datenabgleichs

Ein anderer ebenso wichtiger Baustein des Systems ist der des Datenabgleichs. In diesem Content Management System musste auch eine wichtige Schnittstelle zur Konzernzentrale realisiert werden, die automatisch täglich Daten in diesem System aktualisiert, neu anlegt oder entfernt.
Es war die tägliche Aufgabe der Marketing Abteilung Anpassungen an Daten vorzunehmen um die Marktkonformität zu erhalten. Diese Anpassungen lösten im Hintergrund einen automatischen Datentransfer aus, der des Nächtens via ftp auf den Server übertragen wurde. In den frühen Morgenstunden wurden dann diese in Textform abgelegten Daten via cron Job aufgerufen und mit einer Java Applikation abgearbeitet.
Die Java Datei verantwortlich für den Datenabgleich im Auszug ist im Anhang III, Listung 3.4: Java Code für Datenabgleich zu finden.

Der Zweck dieser Applikation war es die nächtens erhaltenen Daten (welche einen neueren Stand hatten) gegen die derzeit im Internet online gestellten Daten zu ersetzten.
In diesem Java Programm werden die insgesamt 38 Textdateien nach einer bestimmten Ordnung aufgerufen (da einige Daten zuerst eingelesen werden mussten um in anderen Dateien weiterlesen zu können) Es wurden die Daten aus den Textdateien direkt in die MySql Datenbank zur weiteren Verarbeitung importiert. Die Daten umfassten Marken, Modelle, technische Daten, Ausstattungen und Versionen der Fahrzeuge.
Die Veränderungen durch den Datenabgleich wurden in einer weiteren Datenbank abgelegt, um die Änderungen für die User nachvollziehbar machen zu können. In diesen täglichen Updates und in den Aufbereitungen der Daten war eine große Gefahr von Fehlern und falsch abgelegten Daten verborgen, was mich dann zur Änderung der Funktionalität brachte. Wir bauten einfach einen Zwischenschritt in der Datenübernahme ein: in der Praxis erhielten wir täglich einen Datensatz im Text Format von der Zentrale, dieser wurde weiterhin abgearbeitet jedoch nur in eine Zwischeninstanz der Datenbank abgelegt. Nach Kontrolle dieser Daten durch die Verantwortlichen im Marketing wurde durch manuelles auslösen die Daten mit den online Daten aktualisiert.
Hiermit konnte ich in der Vorimplementierungsphase bereits ein großes Datenrisiko gegenüber Dritten (den Usern im World Wide Web, die aufgrund dieser Daten sich eine Konfiguration dieses Produktes erstellen konnten, und dies doch eine gute Qualität der Daten gewährleisten sollte) durch die eingebaute Kontrolle vermeiden.

4. Testen

4.1. Evaluierung der Systemfunktionalität

Die Systeme werden vor dem ersten Einsatz in einem virtuellen Netzwerk online gehen. Dieses Netzwerk bestehend aus dem Server und 5 Clients wird auf ihre Funktionalität getestet.(Tests von Apache, ftp, Mailserver, etc.) Ich simuliere die Zugriffe auf die Web Site in diesem virtuellen Netzwerk.
In einem zweiten Schritt wird auch durch ein Simulationstool Last simulieren und auch die Performance des Systems.

Die Tools die von Linux Standard Komponenten abgedeckt werden sind die

- Last der CPU
- Analyse der Daten die effektiv über das LAN geschickt werden

Im Rahmen dieser Evaluierung baue ich den Server in einem Testlabor auf. Die einzige aktive Komponente ist ein Switch, der den Server mit den 5 Testclients (Windows XP Maschinen) verbindet. Nach dem Start des Servers werden die Monitoring Tools aktiviert um sowohl den Traffic über den eth1 Port zu überwachen als auch die Last der CPU. Ein Tool zur Analyse des Logs von Apache wird ebenfalls installiert um die Log Auswertungen mit in den Test einfließen lassen zu können.

Der Kern des Tests besteht aber aus dem Lasten Test, wo hier das Tool der Firma Mercury zum Einsatz kam.
Es wird ein proaktiver Ansatz gewählt um Software und seine Performance zu testen. Da diese Software in einem Web Umfeld eingesetzt wird soll diese auch den dort nötigen Anforderungen gerecht werden und ausreichend skaliert sein.

Eine Studie der Firma Mercury belegt, dass die Unternehmen Ihre Software Applikationen öfter testen, was auf die gewünschte Verlässlichkeit von Software schließen lässt. Es soll das Risiko der Unternehmen reduziert werden und die Performance der Software daraus gesteigert werden. [HpMer3]

QUESTION	OPTION	% RESPONSE
What are you load testing?	New application development	85%
(Select all that apply)	Major Upgrades	72%
	Patches/Service Packs	40%
	Integration Initiatives	39%
	Other	19%
When do you load test?	When the app is in development	40%
(Select all that apply)	When the app gets to QA	69%
	Right before release	60%
	In staging	38%
	In Production	35%

Abbildung 4.1: Ergebnisse der Survey

Resumeé [HpMer3]:

71 % der Unternehmen gaben an Ihre Software öfters zu testen als früher

85% der Unternehmen testen neue Applikationen

72 % testen grössere Updates

nur 40% testen patches und Service Pack upgrades

Der grafische Aufbau des Performance Centers [HpMer3]:

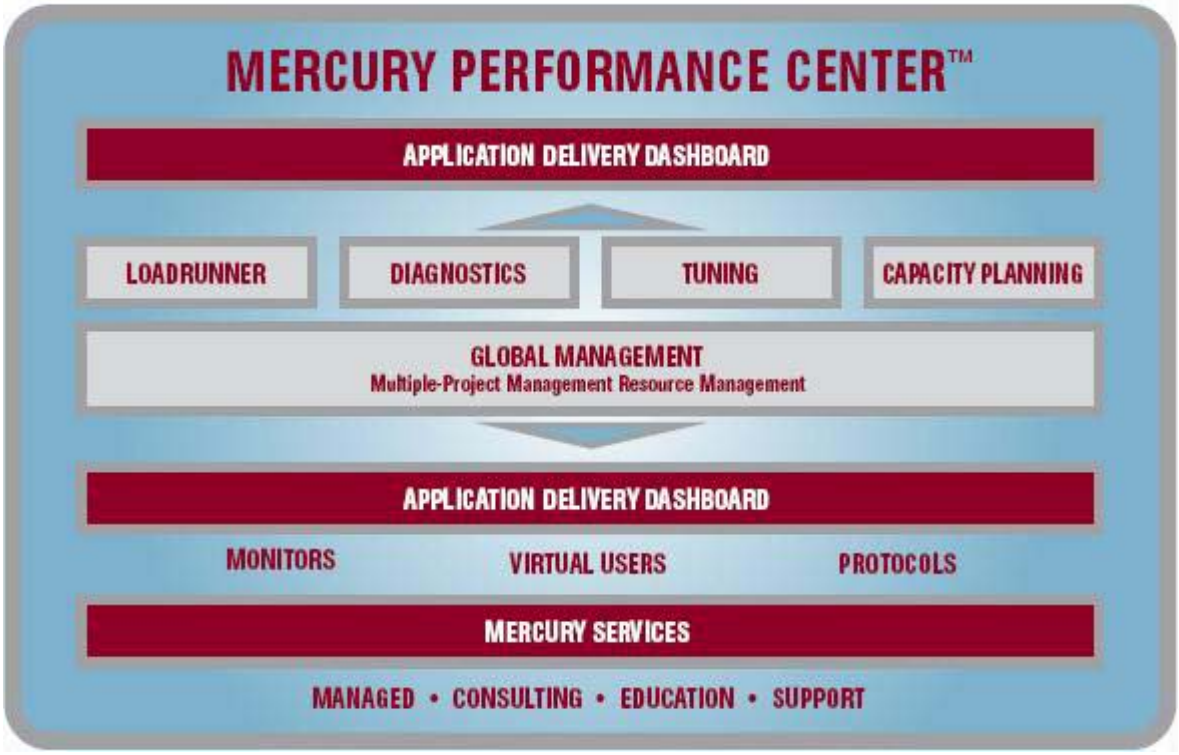


Abbildung 4.2: Die Komponenten des Mercury Performance Centers

4.2. Funktionen des Mercury Load Runner

Der Mercury LoadRunner stellt einen Industriestandard in Performance testing dar. Es ist ein Produkt welches im Mercury Performance Center integriert ist und die Skalierbarkeit, das Verhalten und die Performance testet. Es kann tausende von virtuellen Usern emulieren, es kann Flaschenhälse identifizieren und isolieren und auf Tier Basis feststellen ob eine Applikation gute Performance hat und ausreichend skaliert ist [HpMer2].

Der LoadRunner sammelt System Level Informationen und auch Informationen von Komponenten durch ein dichtes Array von System Monitoren und Diagnostik Modulen. Diese Diagnostik Module ermöglichen nicht nur Qualitätssicherung Teams Probleme bis in aller Tiefe zu analysieren und auch zu behandeln bevor eine Software online geht. Auch sollen Tuning Maßnahmen ermöglicht werden.

Applikations Load testen ermöglicht Entwicklern Bottlenecks in jeder Komponente der Infrastruktur festzustellen [HpMer1].
2 gängige Methoden für die Implementierung dieses Prozesses sind manuelle und automatisierte Tests. Manuelle Tests jedoch haben sehr viele Schwierigkeiten zu bewältigen um z.B. folgende Problemstellungen zu lösen:

- 100.000 User manuell zu emulieren die mit der Applikation interagieren um System Load zu generieren
- Die Operationen von Usern zu koordinieren
- Response Zeiten zu messen
- Test zu wiederholen (in gleicher Art und Weise)
- Resultate zu vergleichen

Da Load Tests im Normalfall iterativ sind, sollte man Performance Probleme erkennen, das System tunen oder verbessern und das System nochmals testen um eine positive Entwicklung festzustellen – unendlich oft, wenn möglich. Aus diesem Grund ist manuelles testen nicht der richtige Ansatz.

Mit automatisierten Load Testing Tools, können Tests wiederkehrend durchgeführt werden und Resultate automatisch gemessen werden. Durch diesen Umstand sind solche Testing Tools wesentlich kosteneffizienter und sie schließen das Risiko eines menschlichen Fehlers aus.

Heutzutage ist ein automatisierter Test die bevorzugte Wahl um eine Web Applikation zu testen. Die Test Tools verwenden typischerweise 3 Hauptkomponenten um einen Test durchzuführen. Diese beinhalten:

- Eine Kontroll-Konsole, welche die Größe, Anzahl der Läufe und die Skalierung übernimmt
- Virtuelle User, welche Prozesse sind die verwendet werden um den echten User zu simulieren, der einen Prozess ausführt
- Load Server, die die virtuellen Server laufen lassen

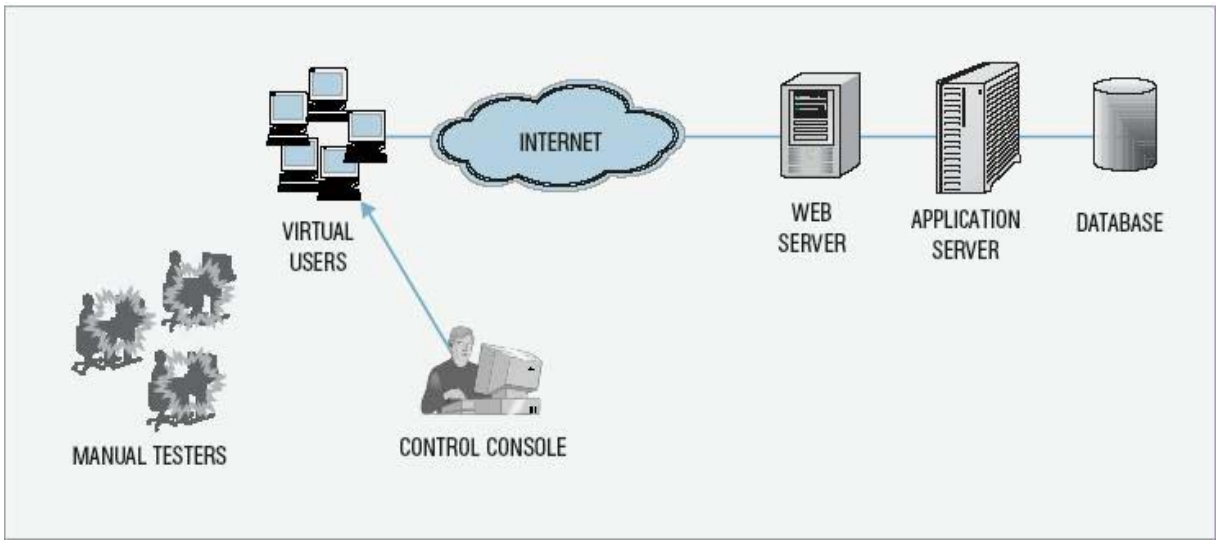


Abbildung 4.3: eine Kontroll Konsole simuliert tausende User und ersetzt so manuelle Tester [HpMer1]

Durch das Verwenden dieser Komponenten können automatisierte Load Testing Tools:

- Manuelle Test User durch virtuelle User ersetzen
- Gleichzeitig viele virtuelle User gegen eine einzelne Maschine testen
- Automatisch die Transaktions Antwortzeit messen
- Einfach Load Szenarien genauso wieder ablaufen lassen um Veränderungen zu sehen

Die Wichtigkeit von automatisierten Test Tools zeigt diese Grafik

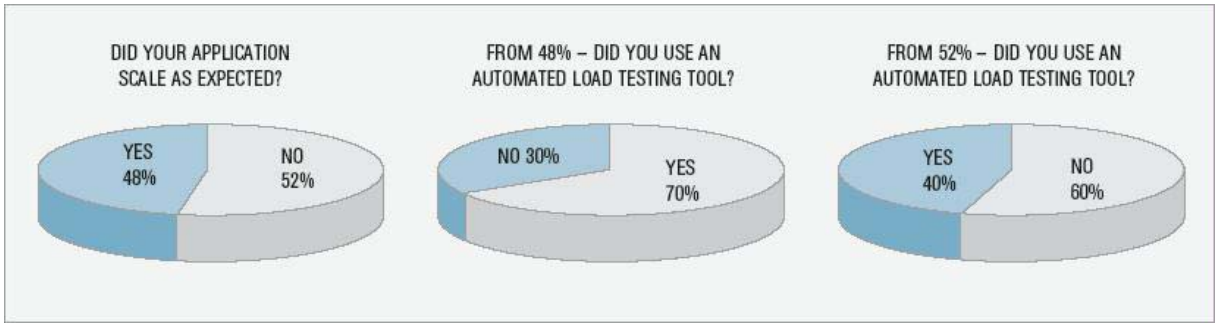
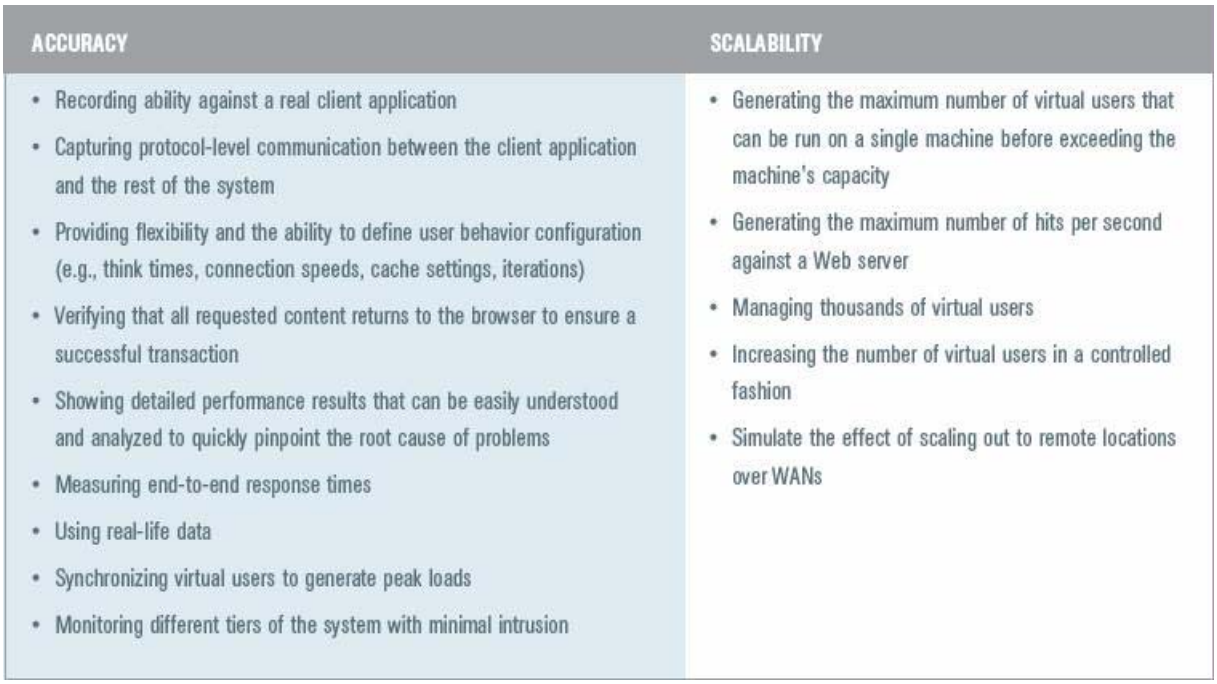


Abbildung 4.4: automatisierte Load Test erleichtern Skalierbarkeit [HpMer1]

Die Key Attribute von Load Testing sind Accuracy und Scalability. Deren Sinn zeigt die nachfolgende Grafik [HpMer1]:



ACCURACY	SCALABILITY
• Recording ability against a real client application • Capturing protocol-level communication between the client application and the rest of the system • Providing flexibility and the ability to define user behavior configuration (e.g., think times, connection speeds, cache settings, iterations) • Verifying that all requested content returns to the browser to ensure a successful transaction • Showing detailed performance results that can be easily understood and analyzed to quickly pinpoint the root cause of problems • Measuring end-to-end response times • Using real-life data • Synchronizing virtual users to generate peak loads • Monitoring different tiers of the system with minimal intrusion	• Generating the maximum number of virtual users that can be run on a single machine before exceeding the machine's capacity • Generating the maximum number of hits per second against a Web server • Managing thousands of virtual users • Increasing the number of virtual users in a controlled fashion • Simulate the effect of scaling out to remote locations over WANs

Abbildung 4.5: Accuracy und Scalability sind Key Indikatoren im Load Testing

4.3. Schritte zum Automatisierten Testen

Der Prozess des automatisierten Testens:
Um den Ansatz zum Load testen richtig zu gestalten sollte man seine Resourcen optimieren, die Hardware, Software und Netzwerk Anforderungen skalieren und sich selbst Erwartungen setzen die dann in SLA's (Service Level Agreements) erreicht werden sollen. Die Verlässlichkeit des Prozesses ist die Basis um Änderungen und Anpassungen zu überprüfen.

Die Schritte eines automatisierten Load Testing Prozesses [HpMer1]:

4.3.1. System Analyse

In diesem Schritt sollten alle KPI (key performance indicators) und Objekte festgelegt werden die im nachfolgenden Test behandelt werden sollen. (Welche Teile der System Architektur, die Anzahl der Concurrent Connections oder Hits pro Sekunde man bei seinem Web Server testen möchte) Man sollte auch einzelne Prozesse, wie z.B. in meinem Fall das Suchen eines Händlers in seinem Gebiet

festlegen. Einen solchen Prozess kann man danach auch in einem SLA (Service Level Agreement) abbilden.
Danach werden die Daten definiert die in dem Test via virtuellen User verwendet werden sollen.
Danach wird nunmehr die Test Policy festgelegt: Diese kann sein Load Test, Stress Test und Capacity Test.
Der Load Test wird verwendet um die Applikation gegen eine Anzahl von Usern zu testen. Das Ziel ist zu sehen ob das System bei der Useranzahl noch eine ansprechende Response Zeit bietet.
Der Stress Test soll Auskunft über die Stabilität und Verlässlichkeit einer Applikation geben.
Der Capacity Test legt die maximale User Anzahl fest welche das System verkraftet.

In dieser Phase sollte man auch ein Augenmerk darauf legen wie die System Architektur dargestellt wird:

- Ob man einen Single Server einsetzt oder einen Cluster Server
- Ob man Load Balancer den Maschinen vorsetzt um die Maschinen zu managen
- Eine generelle Festlegung der Server im Netzwerk (in meinem Fall ein Applikationsserver und ein Mailserver)

4.3.2. Erstellen von Virtual User Scripten:

Hier wird ein Script Recorder verwendet um alle Business Prozess Anforderungen in ein Test Skript zu bringen. Der virtuelle User emuliert den realen User und sein Verhalten.

Es ist wichtig den business Prozess vollständig abzubilden um auch nachher die Messpunkte vergleichen zu können.
Ich startete die Virtual User Definitions Umgebung des Load Runners um die Eingaben aufzuzeichnen. Die Aufzeichnung erfolgt via virtuellen Einstieg in die Web Seite und ein einfaches durchklicken durch die gewünschten Menüpunkte in der

Webseite. Nach Beendigung drückt man Stop der Aufnahme und das Szenario wird generiert.
In meinem Fall wurden hier über 2000 Steps mitgeloggt und in einem Record Log verspeichert. Das Log hat folgendes Aussehen (im Auszug, Details siehe Anhang III, Listung 4.6: Record Log im Auszug):

```
[Network Analyzer (1d58: cd8)] ------------------------------------------------
----------------------------------------
[Network Analyzer (1d58: cd8)] Load Network Traffic Analyzers:
[Network Analyzer (1d58: cd8)]     Analyzer Module: WPLUS (value=)
[Network Analyzer (1d58: cd8)]     Analyzer Module: WebBase
(value=GetHttpProtocolAnalyzer:api_http_filter.dll)
[Network Analyzer (1d58: cd8)]     + Network Analyzer: api_http_filter.dll @
GetHttpProtocolAnalyzer Loaded!
[Network Analyzer (1d58:1610)] Address lookup for TOMBIG = 192.22.11.86
[Network Analyzer (1d58:1ca4)] Address lookup for TOMBIG = 192.22.11.86
[Network Analyzer (1d58:1ca4)] Request Connection: Remote Server @
195.110.210.105:80   (Service=)  (Sid=  1)  PROXIED!
[Web Request      (1d58:1610)] "GET /"
[Network Analyzer (1d58:1610)]   (Sid:  1) Client -> Server : 427 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  1) Server -> Client : 293 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  1) Server -> Client : 315 bytes
(Service=HTTP)
[Network Analyzer (1d58:1ca4)] Address lookup for TOMBIG = 192.22.11.86
[Network Analyzer (1d58:1ca4)] Request Connection: Remote Server @
195.110.210.105:8080   (Service=)  (Sid=  2)  PROXIED!
```

Listung 4.6: Record Log im Auszug

Nach der Aufzeichnung wird das erste Replay durchgeführt um festzustellen, ob in der Aufzeichnung Fehler enthalten sind.
In diesem Replay werden zuerst die Correlationen überprüft und danach wird das virtuelle Skript abgearbeitet. Das Skript hat folgendes Aussehen:

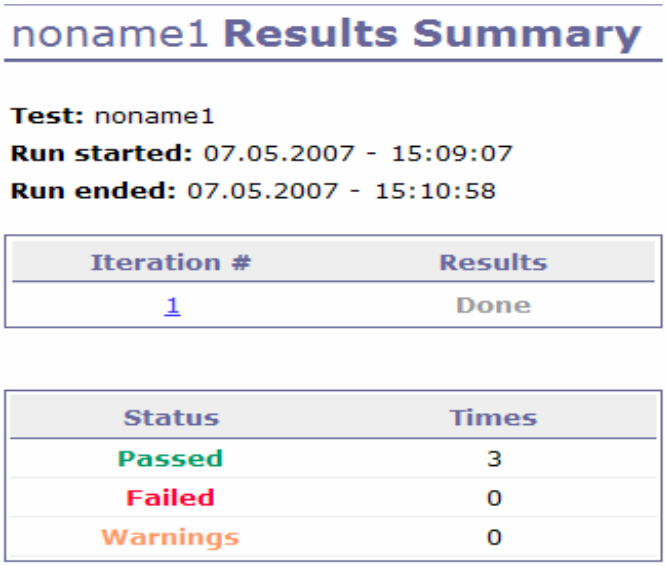
```
vuser_init()
{
        web_url("test.webtom.at",
                "URL=http://test.webtom.at/",
                "Resource=0",
                "RecContentType=text/html",
                "Referer=",
                "Snapshot=t1.inf",
```

```
"Mode=HTML",
EXTRARES,
"Url=http://195.110.210.105:8080/pkw/images/will-over.gif",
"Referer=http://test.webtom.at:8080/pkw/", ENDITEM,
"Url=http://195.110.210.105:8080/pkw/images/habe-over.gif",
"Referer=http://test.webtom.at:8080/pkw/", ENDITEM,
"Url=http://195.110.210.105:8080/pkw/images/alles-over.gif",
"Referer=http://test.webtom.at:8080/pkw/", ENDITEM,
"Url=http://195.110.210.105:8080/pkw/images/partner-over.gif",
"Referer=http://test.webtom.at:8080/pkw/", ENDITEM,
```

Listung 4.7: Script für den Load Test

Zu Beginn wird der Virtuelle User initialisiert. Danach wird der Weg definiert den der virtuelle User gehen soll um die Web Seiten zu testen, in diesem Fall soll er die Seite test.webtom.at aufrufen. Nach erfolgtem Seitenaufruf werden die definierten Items (die bei der Aufzeichnung festgestellt wurden) aufgerufen und ebenfalls geladen. In diesem Beispiel sind es 3 gif Dateien und ein weiterer Link. In weiterer Folge werden so Item nach Item aufgerufen und die Zeit für diese Aufrufe wird von Load Runner aufgezeichnet und protokolliert. Hier im ersten Test der Funktionalität werden nur die Richtigkeit der Links und die Verfügbarkeit der einzelnen Seiten getestet.

Das Ergebnis wird in nachfolgender Grafik dargestellt:



noname1 **Results Summary**

Test: noname1
Run started: 07.05.2007 - 15:09:07
Run ended: 07.05.2007 - 15:10:58

Iteration #	Results
1	Done

Status	Times
Passed	3
Failed	0
Warnings	0

Abbildung 4.8: Resultat des Replay

4.3.3. Festelegen des User Verhaltens

Run-time settings definieren den Weg welchen das Skript nimmt um genau das Verhalten eines echten Users zu simulieren. Settings können Stehzeit, Verbindungsgeschwindigkeit und error Handling festlegen.
Nach der Installation des Tools, begann ich die Schritte für den Systemtest zu definieren. Ich wählte den Einstieg auf der Hauptseite, Navigation zu einem Untermenüpunkt, Auswahl eines weiteren Untermenüpunktes und Anzeige des dort befindlichen Contents.
Danach werden die Load Parameter der Virtuellen User eingestellt. Ich definierte die Parameter wie folgt:

Iterationen pro User 1

Anzahl der Virtuellen User 5 (beide Werte für den ersten Durchlauf)

Vorher definiere ich noch einen besonderen Abschnitt des Tests um von diesem speziellen Abschnitt die Ladezeit der Seite zu bekommen. In LoadRunner wird dies mit einem speziellen Transaction Monitor gemacht. Die Web Seiten werden nach Abschnitten dargestellt und ich kann mit der Maus einen Startpunkt zur Beobachtung

und einen Endpunkt der Beobachtung definieren. Die Grafische Umsetzung sieht dann wie folgt aus:

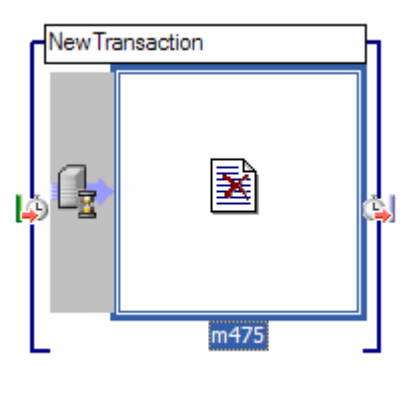


Abbildung 4.9: Transaktionsmonitoring

Am Linken Rand wird der Beginn festgelegt, in der Mitte steht der Namen der Webseite (m475) und danach wir der Beobachtungs Endpunkt festgelegt. Oben wird dann noch eine Name für diese spezielle Transaktion vergeben.

4.3.4. Erstellen des Load Test Scenarios

Das Load Test Szenario beinhaltet Informationen über die Gruppen von virtuellen Usern, welche die Skripts laufen lassen und den Maschinen auf denen die Gruppen laufen.
Es ist wichtig hier alle Definitionen im vor hinein zu machen.
Wir testen in Summe 5 verschiede Szenarien im Detail. Hier erkläre ich die Abläufe anhand des Scenarios1 welches einen Einstieg in die 3. Ebene des CMS Systems simuliert.

4.3.5. Erstellen von Network Impact Tests

In diesen Network Tests werden die Parameter des Netzwerkes verwaltet. Man hat hier die Möglichkeit verschiedene Anbindungen zu testen und zu simulieren (Veränderungen der Bandbreite, Struktur, etc..). Diese Tests können dann

Spezifikationen an das Netzwerk festlegen um der Applikation eine geeignete Infrastruktur zu bieten.

4.3.6. Start des Load Tests Szenarios und Monitoring der Performance

Das Real Time Monitoring erlaubt dem Tester jederzeit während des Laufes den Monitor zu sehen. Jede Komponente braucht Aufmerksamkeit und sollte am Monitor verfolgt werden: die Clients, das Netzwerk, der Web Server, der Applikationsserver, die Datenbank,…
Der Real Time Monitor erlaubt es Flaschenhälse schon während des Tests festzustellen und zu analysieren. Man kann auch die Sicht eines einzelnen Tiers, eines Servers oder einer Komponente während des Tests wählen.
Ich starte den Mercury LoadRunner Controller, der das so genannte und eben definierte Szenario1 ablaufen lässt. Man startet mit „Run Scenario“ die Umgebung und die Zeiten der Antwort der einzelnen Schritte werden in Grafiken gesammelt und extrem übersichtlich dargestellt.

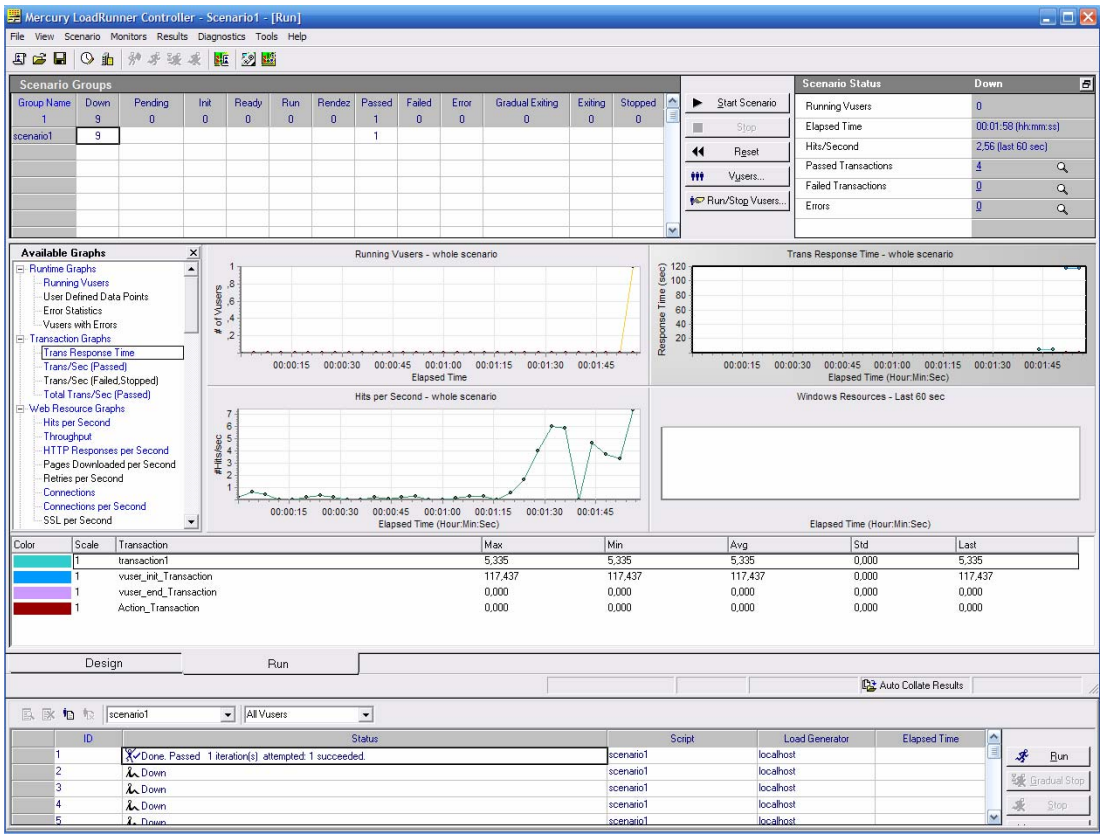


Abbildung 4.10: LoadRunner Controller Gesamtbild

Die Module im einzelnen:

Der Monitor der Transaction die wir im Vorfeld definiert haben. Die Grafik zeigt eine Kurve, die nur den Abschnitt der vordefinierten Transaktion betrifft. Wir sehen das 0,563 sec pro Transaktion verbraucht werden.

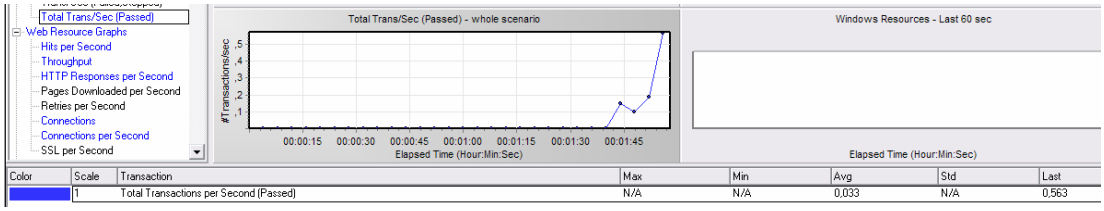


Abbildung 4.11: LoadRunner Transaction Monitor

Der Monitor der http Responses:

Wir sehen hier in der Grafik die Antwortzeiten dargestellt. http_200 hat eine durchschnittliche Antwortzeit von 1,342 sec.

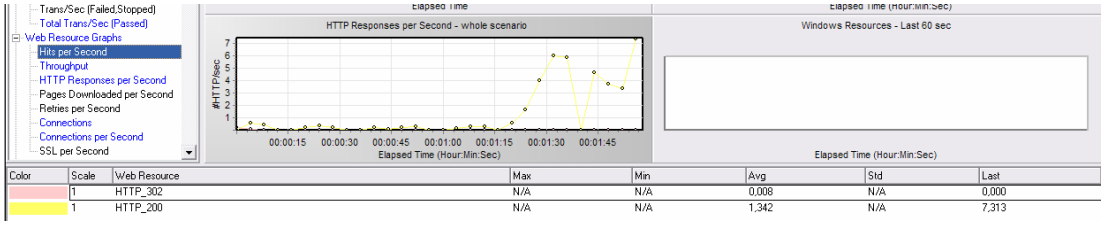


Abbildung 4.12: LoadRunner http Response

Der Monitor des Durchsatzes:

Wir sehen in der Grafik das der Durchsatz im Schnitt 24055 bytes/ sec betrug.

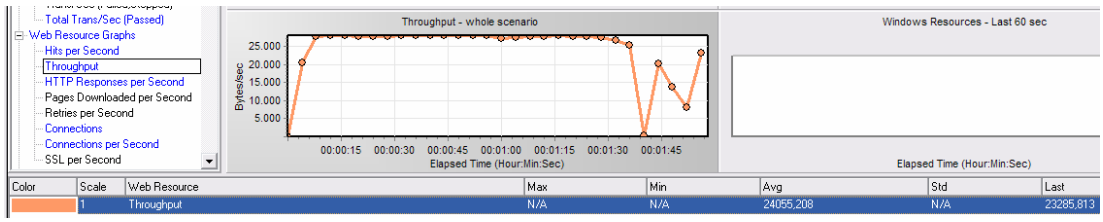


Abbildung 4.13: LoadRunner Durchsatz

4.3.7. Analyse und Auswertung der Ergebnisse

Das ist der eigentlich wichtigste Teil des Tests. Das Ergebnis anzeigen lassen um Performance und Bottlenecks zu analysieren.
Die Auswertung bietet eine Reihe von Grafiken um den end-to-end Test anzuzeigen.

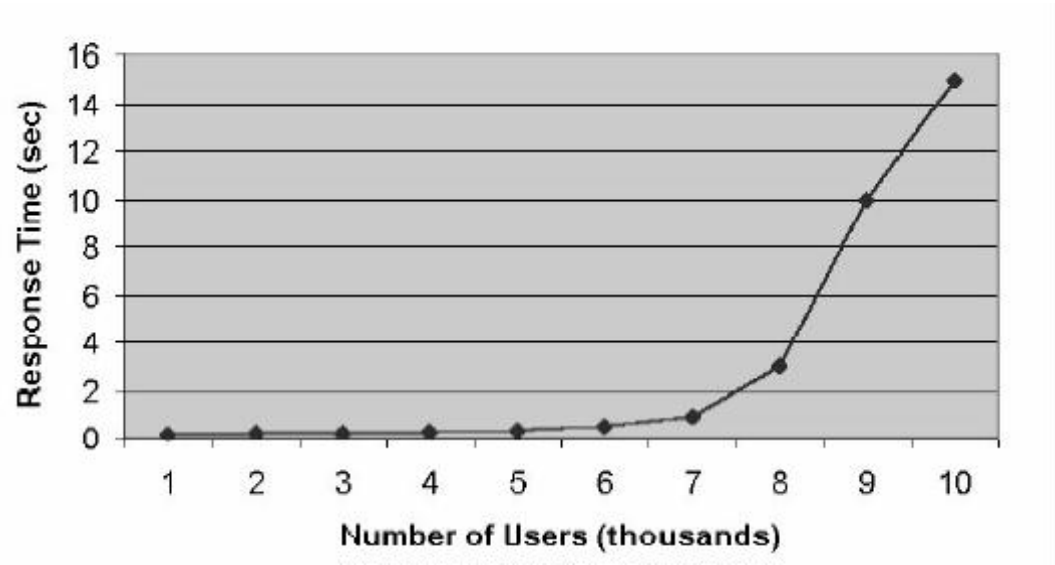


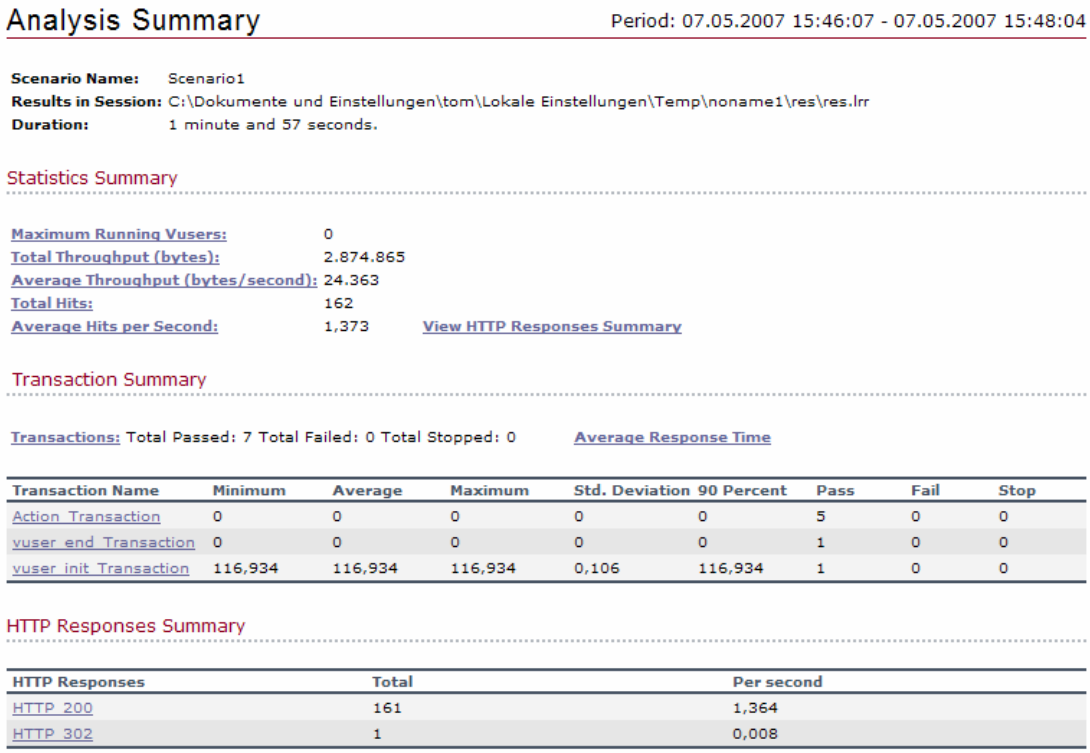
Abbildung 4.14: Userverhalten [HpMer1]

Die obige Tabelle zeigt die totale Anzahl der virtuellen User gegen die Response Zeit. Diese Grafik zeigt den akzeptablen Bereich der Applikation (bei dem die Antwort Zeit noch im grünen Bereich liegt).

Abbildung 4.15: Transaktionen [HpMer1]

Die obige Tabelle zeigt die Anzahl der Transactions die während des Tests gemacht wurden an. Das hilft Bottlenecks aufzuzeigen und die System Performance zu verbessern. Der Test sollte neuerlich durchlaufen werden um eine Änderung zu sehen.

Danach kann man die generierten Daten im Mercury LoadRunner Analysis Tool genau unter die Lupe nehmen und weiter analysieren.
In der nachfolgenden Grafik die Analysis Summary die einen sehr guten Gesamtüberblick über die Leistung des Systems gibt:



Analysis Summary

Period: 07.05.2007 15:46:07 - 07.05.2007 15:48:04

Scenario Name: Scenario1
Results in Session: C:\Dokumente und Einstellungen\tom\Lokale Einstellungen\Temp\noname1\res\res.lrr
Duration: 1 minute and 57 seconds.

Statistics Summary

Maximum Running Vusers:	0
Total Throughput (bytes):	2.874.865
Average Throughput (bytes/second):	24.363
Total Hits:	162
Average Hits per Second:	1,373

View HTTP Responses Summary

Transaction Summary

Transactions: Total Passed: 7 Total Failed: 0 Total Stopped: 0 Average Response Time

Transaction Name	Minimum	Average	Maximum	Std. Deviation	90 Percent	Pass	Fail	Stop
Action_Transaction	0	0	0	0	0	5	0	0
vuser_end_Transaction	0	0	0	0	0	1	0	0
vuser_init_Transaction	116,934	116,934	116,934	0,106	116,934	1	0	0

HTTP Responses Summary

HTTP Responses	Total	Per second
HTTP_200	161	1,364
HTTP_302	1	0,008

Abbildung 4.16: Analysis Summary

4.4. Die Testergebnisse

Nachdem die Seiten mit dem Load Testing Tool geprüft wurden, habe ich mir auch die Serverseitigen Statistiken angesehen, um auch die von Linux zur Verfügung gestellten Tools zur Analyse zu verwenden. Zu diesem Zweck werden die Logs der CPU, des Netzwerktraffic und das Apache Log analysiert.

Nachfoldende Grafiken sollen das Ergebnis des Load Tests Serverseitig noch unterstützen:

Die Auslastung der CPU:

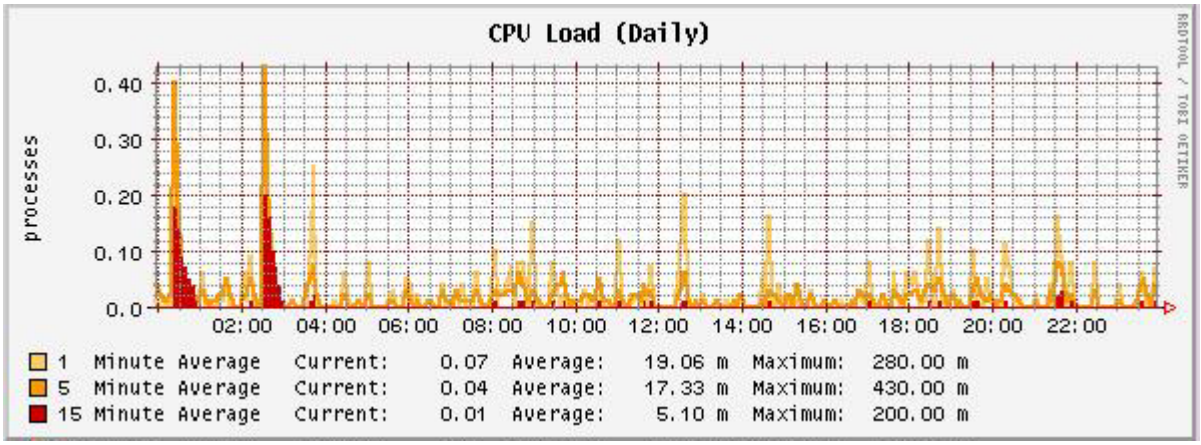


Abbildung 4.17: Grafik der CPU Auslastung

Die Daten des Netzwerktraffics:

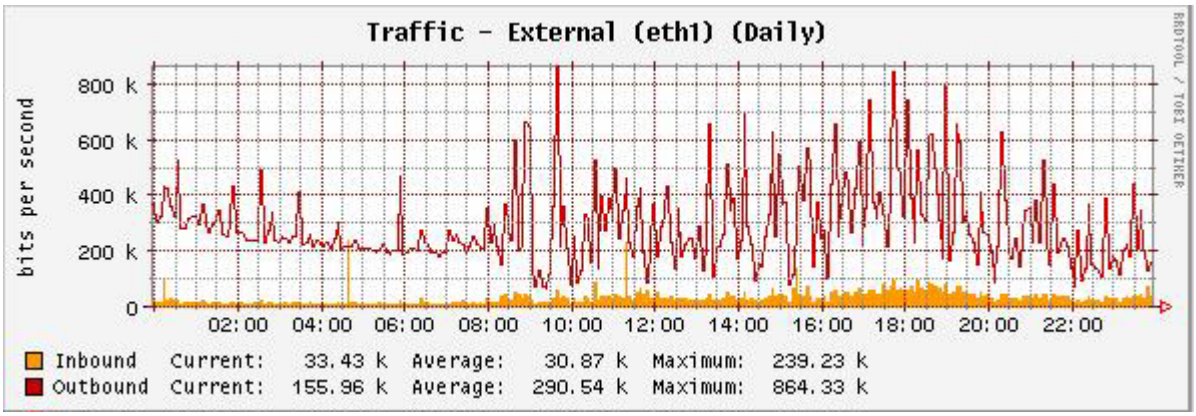


Abbildung 4.18: Grafik des Netzwerk Traffics

Die ersten Tests zeigten eine zufriedenstellende Antwort der Maschine. Die oben gezeigten Graphen lassen genau erkennen, dass das System wenig gefordert wurde und gut skaliert ist.

Im 2. Test wurde das Worst Case Szenario getestet. Ich erstellte Skripte die einen weit größeren Seitenzugriff simulierten.

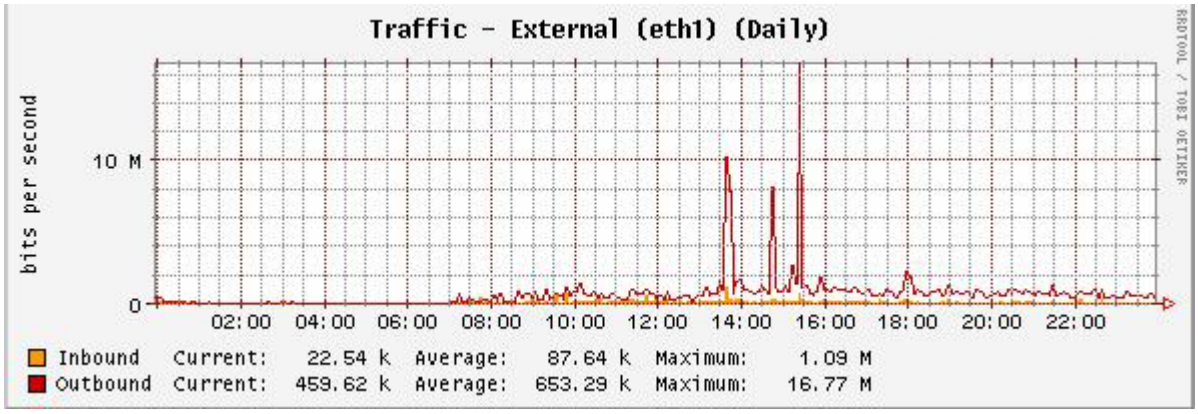


Abbildung 4.19: Grafik des Netzwerk Traffics 2

In der Grafik lassen sich die 3 Starts der Applikationen genau erkennen. Die Simulationen der Seitezugriffe zeigten eine Last von 10 MBit im ersten Start, 8 MBit im 2. Start und 16,77 MBit in der letzten Simulation.
Auch dieses Verhalten zeigte eine gute Skalierung des Environments.

4.5. Content Testing im Pair Testing Verfahren

Hier ging es ans testen des CMS Inhaltes. Im zweiten Schritt wurde sehr genau das User Interface für die Dateneingabe getestet.
Die Erstellung der ersten Testpläne umfassten die Aufstellung der aktuellen Sitemap und die dazugehörigen Links der Seiten.
Die Sitemap umfasste mehr als 90 Referenz Seiten und ihre dazugehörigen Unterseiten. Aufgrund dieser Größe entschied ich mich für das Modell des pair testing um das CMS System und seinen Inhalt auf Vollständigkeit und tote Links zu testen.

Im Ansatz, der im Artikel von Prof. Williams, North Carolina State University Computer Science [Pair1] , beschreiben ist, besteht die Grundidee von Pair Testing darin, das 2 Menschen gemeinsam an z.B. einem Programm arbeiten. Der eine ist der "Fahrer" der andere der "Beobachter". Der Fahrer hat Kontrolle über die Tastatur und die Maus, der Beobachter sieht auf das getippte oder denkt einfach mit. Bei Bedarf können die Rollen getauscht werden. So arbeiten 2 Menschen gleichzeitig am selben Code, Test oder Algorithmus und können so Fehler vermeiden, bevor Sie passieren.

Den selben Ansatz kann man im Artikel "Strengthening the Case for Pair Programming" von Laurie Williams (North Carolina State University), Robert Kessler (University of Utah), Ward Cunningham (Cunningham & Cunningham Inc.) und Ron Jeffries wiederfinden. Auch hier wird eine gesteigerte Effizienz in der Entwicklung von Programmen oder Durchführung von Tests festgestellt.

Es wurden dann in Testpaaren die gesamten Seiten des aufgebauten CMS Systems nach den Prinzipien des pair testings auf deren Vollständigkeit, auf Rechtschreibfehler, auf logische Zusammengehörigkeit und vor allem Richtigkeit der Daten überprüft, die angegebenen Links wurden auf Vorhandensein getestet und vorhandene Grafiken wurden auf Vorhandensein kontrolliert.
Die Ergebnisse wurden in ein Testprotokoll eingetragen und zur Korrektur herangezogen.
Im nächsten Schritt wurde das Interface für die tägliche Datenübernahme kontrolliert. Es wurden Testdaten hergenommen und diese dann via Web Interface auf der aktuellen Seite nach der Übernahme überprüft. Dies wurde ebenso in Testpaaren durchgeführt und erforderte ein hohes Maß an Genauigkeit und Verständnis für die Zahlen in den vorgesehenen Feldern.
Diese Ergebnisse wurden ebenso in einem einfachen Testprotokoll festgehalten und zur Korrektur herangezogen.

5. Transition:

Hier werden einige übergreifende Applikative Tests gemacht wie z.B. bei der Prospektbestellung der Ablauf der eingebundenen Mail Funktion, wie die Archive funktionieren und auch die Performance sich dazu verhält.

5.1. Onlinestellung

Nach erfolgreicher Gestaltung meiner Systemtests wurde das endgültige Zielsystem direkt beim Provider, an seinem finalen Standort, installiert und ins Netz gestellt.
Die Onlinestellung selber war die kleinste Hürde die es zu nehmen galt. Wir nahmen den alten Server offline, installierten an dessen Stelle die neue Maschine und setzten die IP auf die in den bisherigen DNS Servern verspeicherte Adresse, und die Maschine war sofort online.

Die Content Management relevanten Daten und die restliche Systemumgebung wurde vorher vom Testserver übertragen. Danach wurde die Integrität der Daten überprüft und das System wurde komplett gesichert.

Der 2. Server, der als Hot Standby Server konzipiert wurde, wurde dann mit dem erstellten Vollbackup geladen und wurde den erforderlichen Systemtests unterzogen.

5.2. Backup

Nachdem diese Schritte vollzogen waren, wurden die täglichen automatisierten Sicherungen in Form einer Backup to Disk Sicherung in ein anderes geschütztes Filesystem eingerichtet.
In der Crontab exekutiert ein dafür eingerichteter cron Job täglich des Nächtens die Sicherung der Daten auf das andere File System. Es kommt hier das Kopieren mittels scp zum Einsatz. Scp wurde gewählt um auch die einzelnen Berechtigungen der Files mit zu übertragen.
Das 2. wichtige Backup, war das Backup der Daten aus der MySQL Datenbank. Ich verwende ebenso die cron Tabelle um dieses Backup täglich nächtens durchzuführen. Von MySQL wird hierzu ein Perl Programm standardisiert mitgeliefert. In der Cron Tabelle muss man folgenden Eintrag vornehmen, mysql/backup.pl –all, um die gesamten Tabellen die sich in MySQL befinden als sog. Dump File auf die Platte wegzuschreiben. Wöchentlich wird das letzte Dump File genommen und auf dem Cold Standby Server eingespielt und die Daten werden so an einem anderen Ort Ausfallssicher aufbewahrt, so liegt das Risiko der veränderten Daten maximal bei einer Woche.
Das 3. Backup betrifft die Daten die sich in Zope befinden. Da Zope ja wie vorangegangen eine eigene Umgebung ist, müssen auch hier die Daten eigens gesichert werden. Ich habe hierzu 2 Anätze der Sicherung der Daten aus dem Zope System.
Einerseits kommt wieder die crontab ins Spiel, die die Daten täglich wieder auf ein geschütztes Filesystem kopiert. Hier wird das gesamte System von Zope welches sich im Unterverzeichnis var befindet wegkopiert. Die wichtigen Dateien sind hier die Data.fs.* Dateien.

Weiters werden monatliche Zip Files aus der Umgebung selbst erstellt. Zope bietet hierzu die Möglichkeit eines sog. Files Exports an. Man klickt auf das zu exportierende Environment (in meinem Fall eine spezielle Marke) und kann dann den gesamten Inhalt der Marke mit allen für Zope Notwendigen Informationen auf die lokale Platte herunterladen, oder gleich auf den Cold Standby Server zum Import übertragen. Auf dem Cold Standby Server werden die Daten 1x pro Monat via Export und Import aktualisiert und auf Vollständigkeit überprüft.
Die täglichen Log Files wurden in die Cron Tab eingebunden um ein Log Rotate und ebenso eine Sicherung der Logfiles auf ein anderes Filesystem zu enablen.

5.3. Monitoring

Die Überwachungstools wurden gestartet um auch hier wie nachfolgend beschrieben effiziente Monitoring und Analyse Möglichkeit zu haben und das System so wie in den vorangegangenen Tests zu überwachen und die konfigurierten Alarm Mechanismen in Betrieb zu nehmen.

Ziel war es ein effizientes Monitoring von Einzelsystemen und Komponenten zu implementieren um anhand eines zentralen Monitoring Systems agieren zu können.

5.3.1. Nagios

Die von mir gewählte Systemmonitoring-Lösung Nagios [NagDoc1] in Linux deckt folgende, häufig genannte Anforderungen ab:

- Überwachung für die gesamte definierte IT-Infrastruktur
- Zentrales Management
- Einfache Auswertungserstellung inkl. Grafiken
- Report Scheduling
- Langzeitstatistiken, Trendanalyse
- SMS Schnittstelle
- Korrelation von Einzelmessungen zu einem Gesamtzustand
- Einfache Bedienung
- Sichere Remote Administration
- Maßgeschneiderte Implementierung
- Dokumentation von „Zuständen“

· Überwachung von Service Level Agreements
· Zusammengefasste Darstellungen für das Management
· Keine zusätzliche Belastung des Netzwerksverkehrs und Belastung der Zielsysteme

In die Überwachung können folgende Systeme und Netzwerkkomponenten eingebunden werden:

· Microsoft Windows ab Version NT4
· Novell NetWare ab Version 6.0
· Red Hat Linux ab Version 6.2
· SuSE Linux ab Version 7.0
· OpenBSD ab Version 3.2
· SUN Solaris ab Version 5.7
· MAC OS ab Version 9
· FreeBSD, NetBSD, AX oder HP UX sofern rsh oder ssh Zugriff möglich ist
· Cisco Pix Firewall ab Version 5
· Checkpoint Firewall ab NG
· SNMP fähige Netzwerkkomponenten (z.B.: Router, Switches, etc)
· SNMP fähige Endgeräte (z.B.: USV, Temperaturüberwachungssysteme, etc)

Die Kosten für die auf dem „Freeware“ Framework Nagios basierende Lösung sind wesentlich geringer als die Lizenzkosten von Produkten mit vergleichbarer Funktionalität. Künftige Systemerweiterungen innerhalb des Produktportfolios verursachen keine weiteren Kosten.
Die Vorteile und relevanten Entscheidungskriterien sind:

· Nagios ist frei verfügbar und kann bei Bedarf modifiziert werden
· automatische Darstellung der Netzwerkumgebung
· exakte Benachrichtigung aufgrund ausgereifter Eskalationslogik, z.B. wird bei einem Komplettausfall eines Systems genau der jeweilige Dienst gemeldet (und nicht, dass diverse Dienste dieser Maschine nicht erreichbar sind)
· Eskalationen basierend auf der Netzwerktopologie sind möglich
· keine Agenten auf den zu überwachenden Systemen

· keine Kosten für Updates von Agenten auf den Zielsystemen
· Überwachung von vielen Services bis auf Applikationsebene – z.B. die Verfügbarkeit von DNS wird anhand von DNS Abfragen geprüft
· direkte SMS (Mobiltelefon) Integration zur Information von besonders kritischen Ausfällen, konfigurierbar für Störungen außerhalb der Bürozeiten
· Erweiterungen können selbständig durchgeführt werden

Vorteile die für meinen Einsatz entscheidend waren:

· graphische Darstellung der Auslastungskurven
· optimale Planung von Wartungsfenstern möglich
· Trendanalyse
· rechtzeitiges Erkennen von Ressourcen Engpässen - dadurch zeitgerechtes Entgegensteuern möglich
· dadurch Verhinderung von Systemausfällen
· fundierte Planung von Ressourcen möglich
· wertvolle Informationen für die Budgeterstellung

5.3.2. Beispiele aus dem Betrieb

· Laufzeitenmessungen zum Internetprovider (Qualitätsmessung, Überwachen von SLAs)
· CPU, Memory, Diskcounter, Diskkapazität, Errorcounter, Auslastung im Verlauf eines Tages
· Temperaturüberwachung (Verlauf)
· Applikationsspezifische Auswertungen (z.B. WEB Connection, Citrix User, SQL)

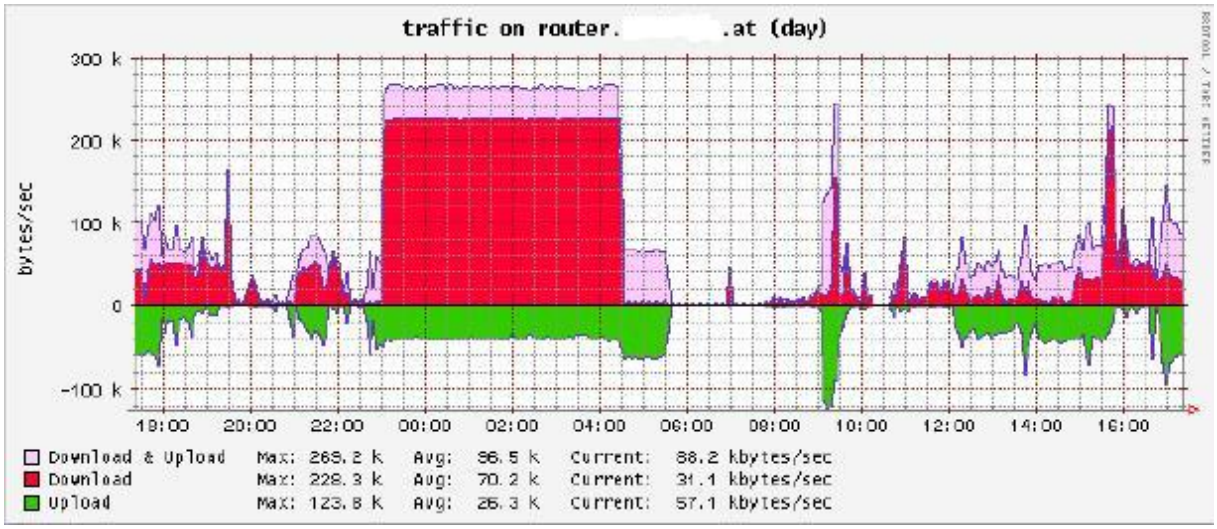


Abbildung 5.1:Grafik des Router Traffics

Beschreibung der Funktionen:

Funktion Langzeitstatistik

Mit dieser Funktionalität ist es möglich, die gesammelten Daten zu analysieren (z.B. in MS Excel) und zum Bespiel für Resourcenplanung einzusetzen.

Date	Time	Weekday	Server	Service	Query	Status #	Status	Text	Correlated State
2005-08-21	01:45:00	Sun	IT-Services	Internet	/var/ /etc/scor/IT-ServicesInternet.txt	0	OK	router .Internet Line Status=OK router .Internet Line Quality=OK	0
2005-09-20	02:25:00	Tue	IT-Services	Internet	/var/ /etc/scor/IT-ServicesInternet.txt	2	CRITICAL	router .Internet Line Status=CRITICAL router .Internet Line Quality=CRITICAL	2
2005-09-20	03:00:00	Tue	IT-Services	Internet	/var/ /etc/scor/IT-ServicesInternet.txt	0	OK	router .Internet Line Status=OK router .Internet Line Quality=OK	0
2005-09-30	12:05:00	Fri	IT-Services	Internet	/var/ 'etc/scor/IT-ServicesInternet.txt	2	CRITICAL	router .Internet Line Status=CRITICAL router .Internet Line Quality=OK	2
2005-09-30	12:10:00	Fri	IT-Services	Internet	/var/ /etc/scor/IT-ServicesInternet.txt	0	OK	router .Internet Line Status=OK router .Internet Line Quality=OK	0
2005-10-04	02:00:00	Tue	IT-Services	Internet	/var/ /etc/scor/IT-ServicesInternet.txt	2	CRITICAL	router .Internet Line Status=OK router .Internet Line Quality=CRITICAL	2
2005-10-04	02:05:00	Tue	IT-Services	Internet	/var/ /etc/scor/IT-ServicesInternet.txt	0	OK	router .Internet Line Status=OK router .Internet Line Quality=OK	0

Abbildung 5.2: Grafik des Statistik Auswertung

Funktion Reporting:

Diese Module ermöglichen das Vergleichen und Bewerten von gesammelten Messdaten. Die Darstellung dieser Reports erfolgt im html und im xls Format. Damit wird uns z.B. ermöglicht, eine Gesamtübersicht der Diskauslastung unserer Systeme zu erhalten.

Vorteile die sich daraus ergeben:

- · Vergleichen von Messdaten
- · Erkennen von Veränderungen– Trends können abgeleitet werden
- · Detailaufzeichnungen ermöglichen eine vorausschauende Planung
- · Bewertung von Auslastungen

Funktion Disk Summary:

Dieses Modul liefert eine Gesamtübersicht der verfügbaren Diskkapazitäten der definierten Maschinen - unabhängig vom Betriebssystem des Zielsystems.

Zum besseren Verständnis, welche Möglichkeiten sich aus dieser Funktionalität ergeben sind in der nachfolgenden Graphik verfügbaren Diskkapazitäten von NetWare Volumes, Unix Partitionen und Windows Disken, automatisch sortiert nach den am meisten „befüllten“ Disken, zu sehen.

Server	Disk / Volume / Partition	Size (MB)	Used (MB)	Free (MB)	Used %	Free %
gho-itd-01	Disk I	416739	372092	44647	89	11
nw-itd-01	Volume KUNDEN	51200	44188	7012	86	14
nw-itd-01	Volume UXKITS	81920	68758	13162	84	16
srv-itd-07	Disk C	9802	8209	1592	84	16
srv-itd-05	Disk D	14331	11873	2458	83	17
srv-itd-02	Disk E - Data2	225286	178059	47228	79	21
nw-itd-01	Volume MSKITS	153600	120498	33103	78	22
nw-itd-01	Volume SYS	7934	6135	1800	77	23
srv-itd-09	Disk C	8375	6433	1942	77	23
srv-itd-09	Disk D	26349	19469	6880	74	26
srv-itd-02	Disk X - Exchange	46077	33525	12552	73	27
nw-itd-02	Volume iFolder	30623	21589	9034	70	30
srv-itd-02	Disk C	15001	10205	4796	68	32

Abbildung 5.3: Grafik der Disk Summary

Die nachfolgende Skizze zeigt schematisch den Weg der Messdaten.

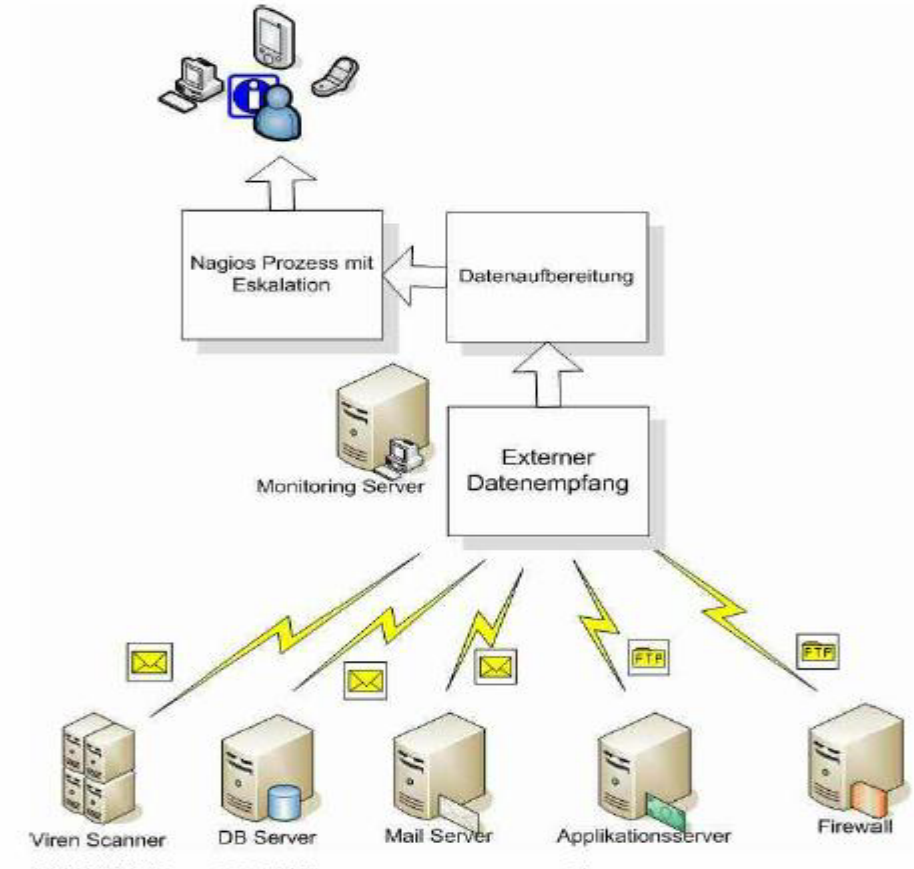


Abbildung 5.4: Weg der Messdaten

Der modulare dreischichtige Aufbau ermöglicht den Empfang beliebiger Messdaten von unterschiedlichsten Plattformen:

· Systemmeldungen bzw. Meldungen von Applikationen können standardisiert
  ins Nagios integriert werden
· viele Applikationen bieten die Möglichkeit, bei Ereignissen
  Benachrichtigungen zu versenden. Mit diesem Modul können diese
  Nachrichten standardisiert übergeben und auf der WEB Oberfläche dargestellt
  werden

Nachfolgende Tabelle zeigt den momentanen Zustand des Logon Services. Der Gesamtzustand wird aus einzelnen Services ermittelt.

Host	Service	Status	Last Check	Last change	Output
srv-itd-02	udp 88 Kerberos	OK	Thu Oct 20 17:39:08 2005	Mon Sep 12 17:35:03 2005	UDP port 88 is open
srv-itd-02	tcp 135 DC	OK	Thu Oct 20 17:38:42 2005	Mon Sep 12 12:38:36 2005	TCP port 135 is open
srv-itd-02	Service NetLogon	OK	Thu Oct 20 17:51:14 2005	Wed Oct 19 23:47:56 2005	Windows service "Netlogon" is running
srv-itd-02	dns query	OK	Thu Oct 20 17:49:51 2005	Thu Oct 20 12:11:52 2005	DNS ok - 1 seconds response time, Address(es) is/are 193.83.153.12
srv-itd-02	host	UP	Thu Oct 20 16:34:44 2005	Thu Oct 20 12:09:32 2005	(Host assumed to be up)
Correlated status		OK	view configuration		

Abbildung 5.5: Grafik des Logon Services

Funktion Netzwerkinterface Table:

Mit dieser Messung lassen sich die Eigenschaften aller Interfaces eines Devices (z.B. CISCO Router, Switch, Windows Server, etc.) übersichtlich darstellen und gleichzeitig messen. Dadurch sind Änderungen in der Konfiguration leicht zu erkennen.

Das folgende Bild zeigt einen Ausschnitt aus der Interfacetabelle - das FastEthernet0/1 Interface wurde Aufgrund eines Leitungsausfalls "down" genommen,

durch die Interface Table Messung erkannt, und die Benachrichtigung der verantwortlichen Administratoren veranlasst.

Name	System Information
ITDESIGN-SDSL	Cisco IOS Software, 1841 Software (C1841-BROADBAND-M), Version 12.4(1b), RELEASE SOFTWARE (fc2),Technical Support: http://www.cisco.com/techsupport,Copyright (c) 1986-2005 by Cisco Systems, Inc.,Compiled Fri 23-Sep-05 19:43 by evmiller

index	Descr	Alias	Errors	AdmnStatus	OperStatus	Speed	IP
1	FastEthernet0/0		1247	up(1)	up(1)	100.00 Mbit	
2	FastEthernet0/1		0	[illegible]	[illegible]	100.00 Mbit	
3	ATM0/0/0		0	up(1)	up(1)	2.05 Mbit	
4	ATM0/1/0		162010	up(1)	up(1)	2.05 Mbit	
5	Null0		0	up(1)	up(1)	4.29 Gbit	

Abbildung 5.6: Grafik der Interfacetable

Weitere Funktionen:

- bei sämtlichen Interfaces wird die Auslastungskurve mitgeschrieben
- Leitungsüberlastungen (z.B. mehr als 80% Auslastung in der letzten Stunde) werden erkannt
- Plug&Play - keine Konfiguration der einzelnen Interfaces erforderlich, alle

Interfaces werden automatisch erkannt

Funktion LogFile Auswertung:
Mit der Loganalyse lassen sich Logfiles unterschiedlicher Quellen automatisiert bewerten.
Als häufiger Anwendungsfall ist hier die Analyse von Backuplogs zu erwähnen: Nach jedem Backup-Job wird das Logfile an die Maschine gesandt und durch, von speziell entwickelten Algorithmen, analysiert. Sollte die Datensicherung nicht erfolgreich sein, werden die verantwortlichen Operatoren benachrichtigt. Das entsprechende Logfile wird archiviert und kann bei Bedarf jederzeit eingesehen werden.
Praktische Beispiele:

- Prüfen von Systemressourcen
- Planen von Systemressourcen
- Abfragen von Applikationszuständen
- Erkennen von Ressourcenengpässen
- Aufzeichnen von Softwareständen
- Erkennen von Sicherheitsschwachstellen

Funktion Mail Service Überwachung:
Dieses Modul überwacht die Laufzeit von Mails durch das Internet. Vom Monitoring Server werden periodisch Mails an einen "echo Account" im Internet abgesetzt und wieder empfangen. Erfolgt der Mailversand über ein bestehendes Mailsystem wie zum Beispiel MS Exchange oder Novell GroupWise, ist dadurch eine lückenlose Überwachung der gesamten Mail-Infrastruktur gegeben.
Vorteile durch diese Überwachung:

- Überblick über Verzögerungszeiten im Internet
- Prüfung der Gateways, Virenscanner etc., die den Mailversand beeinflussen
- Erkennen von Störungen bevor der Benutzter es merkt

Folgendes Beispiel zeigt die Zustelldauer von Mails im Verlauf einer Woche:

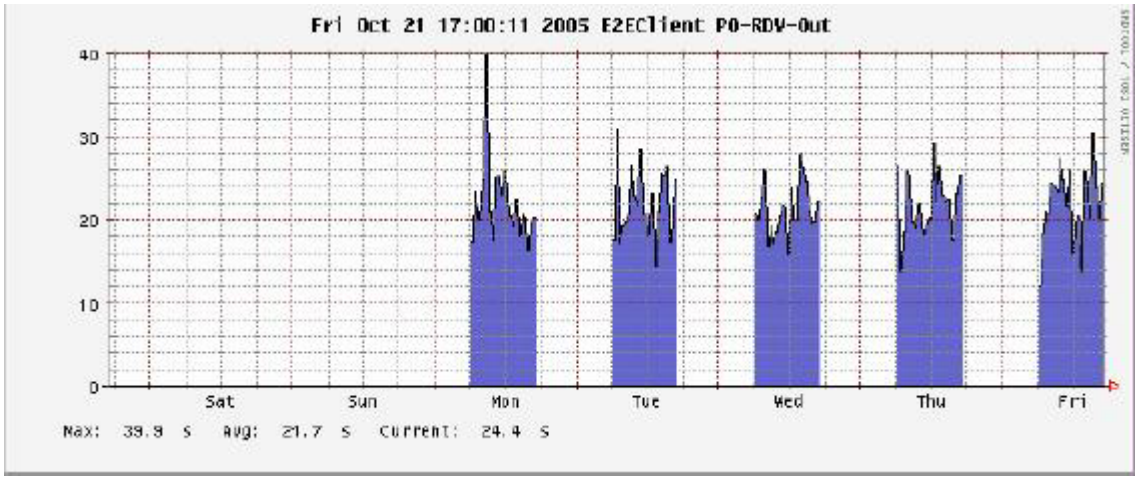


Abbildung 5.7: Grafik des Mail Verlaufes

Zusätzlich zu scriptbasierenden Applikationsmessungen können Transaktionen auch mittels „Applikations-Robotern“ nachgestellt werden. Damit wird ermöglicht, Bedienungsabläufe in Applikationen und Programmen nachzustellen.
Beispiel einer Transaktion:

- Anmelden an einem TerminalServer
- Aufruf einer Datenbank Applikation
- Suchen von bestimmten Datensätzen innerhalb der Applikation

Diese Art von Messungen werden durch den End to End Client unterstützt, erfordern allerdings zusätzliche Software und separate Lizenzen die ich für meine Installation aus Kostengründen nicht erwogen habe.

5.4. Echtbetrieb

Der letzte Schritt in dieser mehrere Monate andauernden Arbeit war der Echtbetrieb des Systemes. Die einzelnen Komponenten der Hardware und Software waren ja schon getestet und machten auch im Echtbetrieb keine Probleme. Ich habe auch noch die vom Betriebssystem angebotenen automatischen Security Patch Updates enabled um für das System auch hinter der Firewall alle Sicherheitsrelevanten Maßnahmen getan zu haben um nicht Sicherheitslücken entstehen zu lassen.
Die Überprüfung der einzelnen Sicherungen auf den Sicherungs-Filesystemen war auch zufrieden stellend verlaufen. Alle Daten waren vollständig, lesbar und konnten auf einem anderen System wiederhergestellt werden.

Das Monitoring der ersten Stunden und Tage verlief auch sehr gut.
Wir konnten dank des Monitoring im Echtbetrieb auch verlässliche Zahlen für das Marketing liefern um zugekaufte Werbung für unsere Homepage auch so bewerten zu können, was einen zusätzlichen Benefit der neuen Struktur brachte.
Ich konnte feststellen, das sich die Load Tests und die vorab geleisteten Test auch in unter Realen Lasten so verhielten wie in den Tools angegeben und simuliert. Das war für mich eine sehr positive Erfahrung. Die User waren auch mit den gebotenen Möglichkeiten der Eingaben und der Pflege der Seiten sehr zufrieden. Auch die Einbindung in weiterer Folge von externen Agenturen konnte sehr einfach und erfolgreich gestaltet werden.
Ich war mit allen Komponenten und ausgewählten Systemen sehr zufrieden und das zu diesem Zeitpunkt skalierte System läuft auch heute noch in dieser Umgebung und Konstellation einwandfrei.

5.5. Conclusions

Die beschriebene Arbeit stellte eine große Herausforderung an mich dar. Die Schwierigkeit lag in der Lösungsfindung für die Fülle von Anforderungen die an mich gestellt wurden.
Ich musste eine komplett neue technische Lösung implementieren, die von der Auswahl des Betriebssystems, der Auswahl der Hardware bis hin zur Festlegung der Softwarekomponenten reichte.
Ein wichtiger Punkt war für mich auch die volle Einbeziehung der User in die Modellierung der einzelnen Komponenten, da ich hier sehr stark auf das nicht IT geschulte Personal Rücksicht nehmen musste. So erlangte ich mehr Verständnis für die Sichtweise des Users.
Ein interessanter Kern meiner Arbeit war auch das Testen der Software, die eigens für diese Anforderung geschrieben wurde. Der Umgang mit dem Testtool, und die am Anfang bestandene Skepsis konnte bald durch die Ergebnisse und sehr professionelle Umgebung der Software ausgeräumt werden. Es war für mich eine große Erfahrung Software anhand eines speziellen Analyse Tools zu skalieren und die Auswertungen dieses Tools in die Implementierung einfließen zu lassen.

Ich stellte auch in diesem großen Projekt fest wie wichtig es war ein gutes Projektmanagement zu haben und auch so die User, externen Programmierer und letztendlich die internen Resourcen planen zu können.
Mit großer Sorgfalt legte ich auch die Strategie der Backups an. Hier konnte ich die Vielfalt der Möglichkeiten von Linux kennen lernen. Einerseits die gebotenen Möglichkeiten von mysql, als auch die Alternativen für die Datensicherung in traditioneller Form.
Eine Schwierigkeit war auch die Budget Vorgabe des Konzerns und der daraus notwendige Einsatz von vielen freien Software Programmen die keine Lizenzkosten verursachten. Es war hier wichtig Augenmerk auf die vielen versteckten Features und die daraus resultierenden Sicherheitsrisiken abzuwägen und sich für eine für unsere Bedürfnisse anpassbare Variante zu entscheiden.
Abschließend freut es mich besonders, dass das vor mehreren Jahren aufgestellte Konzept auch heute noch seine Gültigkeit hat und die seinerzeit erarbeiteten Skalierungen noch heute in der gleichen Form online sind und es lediglich einen durch die Jahre bedingten Hardware Tausch gegeben hat.

ANHANG I :

Liste der Abbildungen:

ANHANG I I:

Literaturverzeichnis:

[Zope25] The Zope Book (2.5 Edition), Online Version, Link besucht am 10.3.2007 unter http://www.zope.org/Documentation/Books/ZopeBook/2_5_edition/

[UnLi1] United Linux Technical Whitepaper, pdf File, download 10.11.2006 von www.unitedlinux.com

[UnLi2] UnitedLinux Version 1.0 Data Sheet, pdf File, download 10.11.2006 von www.unitedlinux.com

[SuseLi1] SUSE® LINUX Professional 9.2 von Novell® , download 10.11.2006 von http://www.novell.com/de-de/products/linuxprofessional/overview.html

[Mysql1] MySQL® Business White Paper, January 31, 2005, Copyright © 2005, MySQL AB

[NagDoc1] Nagios Version 3.x Documentation, Online Version, http://www.nagios.org
Copyright © 1999-2007 Ethan Galstad, Last Updated: 03-20-2007

[HpMer1] White Paper Load Testing to predict Web Performance, download 8.2.2007 von http://downloads.mercury.com/cgi-bin/portal/download/index.jsp

[HpMer1] White Paper Mercury Performance Center Implementation Service Best Practices, download 8.2.2007 von http://downloads.mercury.com/cgi-bin/portal/download/index.jsp

[HpMer1] White Paper Application Delivery Business Driver Survey Results , download 8.2.2007 von http://downloads.mercury.com/cgi-bin/portal/download/index.jsp

[Pair1] Artikel Strengthening the Case for Pair-Programming, Williams, Kessler, Cunningham, Jeffries, besucht am 15.11.2006 http://www.agilesweden.org/doc/StrengtheningTheCaseForPairProgramming.pdf

[Css1]. Historisches Seminar, CSS Cascading Style Sheets, besucht am 10.12.2006, http://www.geschichte.uni-muenchen.de/imp/doku_css.shtml

ANHANG III:

Listungen:

Listung: 2.0: Stylesheet für Content Modul

```
body {margin:0px;padding:0px;
      background-color:#dadccf;
      background-image:url(images/bg_lines.gif)}
img, table {border:none;margin:0px;padding:0px}
p, td, .sm_subnav, .sm_subsubnav, .nav {font-
family:Arial;font-size:12px;font-weight:normal;text-
decoration:none;margin:0px;padding:0px}
.nav { font-size: 11px; color: white; }
/*#head */
#head { /*background:url(images/head.gif);background-
repeat:no-repeat;*/ height:82px;}

#logo {position:absolute; top:23px; left: 45px; }
#metanav {position:absolute;top:30px;left:750px}
#metanav td {padding:0px 5px 0px 5px;
             border-right: 1px solid #5c5c5c;}
.metanav {font-family:Verdana,Arial;font-size:13px;font-
weight:bold;color:#5c5c5c;text-decoration:none}
#mainnav {position:absolute; top:63px; left:181px; z-
index:100;
          border-bottom: 1px solid white;}
#mainnav p {white-space:nowrap}
#admin {position:absolute;top:10px;left:250px}
/*content-table*/
#cont {width:100%; table-layout:fixed;}
/*td linke contentspalte*/
#left {white-space:nowrap; overflow:hidden;
       width:182px; vertical-align:top;
       background-color:<dtml-var sectionColor
missing="#dadccf">;
       background:url(images/background_left.jpg);
       background-repeat:no-repeat;
}


#left_autoexp {white-space:nowrap; overflow:hidden;
       width:182px; vertical-align:top;
       background-color:<dtml-var sectionColor
missing="#dadccf">;
       background:url(images/background_left_autoexpert.jpg);
       background-repeat:no-repeat;
}

#caption_box {width:632px; height:70px;
              background-color: #eaebe3;
              vertical-align: middle;}
#caption {font-size:18px; width: 260px; padding-left:24px;}
#link_configurator { padding-left:0px; font-size:14px; }
#link_configurator a:link { color: #606050; }
#link_configurator a:visited { color: #606050; }
```

```
#right {vertical-align:top}
/* linke layer */
#title {position:absolute;
        top: 350px; left: 0px;
        height:60px;
        padding-left:30px;
        padding-top:0px;
        overflow:hidden;}
#subNav {position:absolute; top: 420px; left: 30px; width:
151px;
         height:225px;
         padding:0px;
}
/* content weite */
#ct, #ctw, #ct table, #ctw table {width:632px}
#ct p, #ct a {color:#000000}
#ctw p, #ctw a {color:#000000}
/* content-bgs */
#ctbg-odd-small, #ctbg-odd-big, #ctbg-even-small, #ctbg-even-
big {width:632px; clear:both}
#ctbg-odd-small, #ctbg-odd-big {background-color: #eaebe3;}
#ctbg-even-small, #ctbg-even-big {background-color: #cfcdbe;}
/* content images (269px und 131px breit)*/
#ctbg-odd-small img, #ctbg-odd-big img {float:right}
#ctbg-even-small img, #ctbg-even-big img {float:left}
/* paddings bei odd und even */
#ctbg-odd-small p {padding:8px 141px 8px 8px}
#ctbg-odd-big p {padding:8px 272px 8px 8px}
#ctbg-even-small p {padding:8px 0px 8px 141px}
#ctbg-even-big p {padding:8px 0px 8px 278px}


/* Angebote Contents */
#ctAngebot, #ctAngebot table {width:632px; margin-bottom:
15px;}
.angebot_image_hoch {float: right;
                     margin-bottom: 8px;
                     margin-left: 16px;}
.angebot_image_quer {padding: 0px 0px 8px 24px;
                     text-align: left;
                     margin-bottom: 8px; }
.angebot_image_quer img {margin-bottom: 5px;}
#ctAngebot p  { padding-left: 24px;
                padding-right: 0px; }
#ctAngebot em { font-size: 12pt;
                font-weight: bold;
                font-style: normal;}
#ctAngebot a { color: black; }

/*links in subNav*/
.sub_nav {background-color: #cfcdbe;
          height: 18px;
          margin-top: 2px;
          padding: 4px 5px 0px 6px;
          cursor: pointer;
          cursor: hand; }
.sub_nav a {text-decoration:none;
            color:#625050; }
.sub_nav_bold {background-color: #606050;
               height: 18px;
```

```
                margin-top: 2px;
                padding: 4px 5px 0px 6px;
                cursor: pointer;
                cursor: hand; }
.sub_nav_bold a {text-decoration:none;
                  font-weight:800;
                  color:#625050; }

/* sitemap */
.sm_subnav, .sm_subsubnav { font-family:Verdana,Arial;
                            font-weight: normal;
                            text-decoration: none;
                            }
.sm_subnav { font-size: 12px; color: white;}
.sm_subsubnav { font-size: 10px; color: #625050;
                margin-left:5px; }

/* tables */
.lines, .spacer {background-color:#dadccf}
.thl {padding:0px 8px 0px 8px;
      font-weight:800; height:30px;
      background-color:#606050;
      color:white; }
.todd, .teven {padding:0px 8px 0px 8px}
.todd {background-color:#cfcdbe}
.teven {background-color:#cfcdbe}
.thlspacer {background-color:<dtml-var sectionColor
missing="#bcc3cd">;width:1px}
.lines {width:1px}
.spacer {height:1px}
/*checkboxn, select, input in forms*/
.sel {border:1px solid black;width:75px}
.box {border:1px solid black;width:250px}
.box1 {border:1px solid black;width:50px}
.box2 {border:1px solid black;width:150px}
/* news box */
#newstable { padding: 0px; width: 625px; }
.edge_top { border-top: 1px solid #A2A29a; }
.edge_bottom { border-bottom: 1px solid #A2A29a; }
.edge_left { border-left: 1px solid #A2A29a; }
.edge_right { border-right: 1px solid #A2A29a; }
.dots { background-image: url(images/dots.png); background-
repeat: repeat-x;
        height: 1px; }
```

Listung 3.2: Code des Newsletter im Auszug

```
<script language="JavaScript" type="text/JavaScript">
<!--
function MM_findObj(n, d) { //v4.01
  var p,i,x;  if(!d) d=document;
if((p=n.indexOf("?"))>0&&parent.frames.length) {
    d=parent.frames[n.substring(p+1)].document;
n=n.substring(0,p);}
  if(!(x=d[n])&&d.all) x=d.all[n]; for
(i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
  for(i=0;!x&&d.layers&&i<d.layers.length;i++)
x=MM_findObj(n,d.layers[i].document);
  if(!x && d.getElementById) x=d.getElementById(n); return x;
}
function MM_validateForm() { //v4.0
  var
i,p,q,nm,test,num,min,max,errors='',args=MM_validateForm.argum
ents;
  for (i=0; i<(args.length-2); i+=3) { test=args[i+2];
val=MM_findObj(args[i]);
    if (val) { nm=val.name; if ((val=val.value)!="") {
      if (test.indexOf('isEmail')!=-1) { p=val.indexOf('@');
        if (p<1 || p==(val.length-1)) errors+='- '+nm+' muss
eine gültige E-mail Addresse enthalten.\n';
      } else if (test!='R') { num = parseFloat(val);
        if (isNaN(val)) errors+='- '+nm+' muss eine Zahl
sein.\n';
        if (test.indexOf('inRange') != -1) {
p=test.indexOf(':');
          min=test.substring(8,p); max=test.substring(p+1);
          if (num<min || max<num) errors+='- '+nm+' must
contain a number between '+min+' and '+max+'.\n';
    } } } else if (test.charAt(0) == 'R') errors += '- '+nm+'
ist notwendig.\n'; }
  } if (errors) alert('Der folgende Fehler ist
aufgetreten:\n'+errors);
  document.MM_returnValue = (errors == '');
}
//-->
</script>
<dtml-if "REQUEST.has_key('send_newsletter')">
<dtml-var "_.render(static_newsletter_send)" missing="??">
<dtml-else>
<form name="newsletter" method="post" action=""
onSubmit="MM_validateForm('vorname','','R','nachname','','R','
email','','RisEmail','plz','','NisNum');return
document.MM_returnValue">
<input type="hidden" name="send_newsletter" value="1">
<table cellpadding=0 cellspacing=2 border=0 width=500>
<tr>
<td colspan=2> </td>
</tr>
<tr>
<td width=137><P align=right><b>Anrede:</b></P></td>
<td><select name="anrede" class=sel>
    <option value=""> </option>
    <option value="Frau">Frau</option>
    <option value="Herr">Herr</option>
```

```
    </select></td>
</tr>
<tr>
<td><P align=right><b>Vorname:</b></P></td>
<td><INPUT TYPE=TEXT NAME="vorname" SIZE=30 class=box></td>
</tr>
<tr>
<td><P align=right><b>Nachname:</b></P></td>
<td><INPUT TYPE=TEXT NAME="nachname" SIZE=30 class=box></td>
</tr>
<tr>
<td><P align=right><b>E-Mail:</b></P></td>
<td><INPUT TYPE=TEXT NAME="email" SIZE=30 class=box></td>
</tr>
<tr>
<td colspan=2> </td>
</tr>
<tr>
<td>
<P align=right>Optionale Angaben:</p>
</td>
<td> </td>
</tr>
<tr>
<td colspan=2> </td>
</tr>
<tr>
<td><P align=right><b>Firma:</b></P></td>
<td><INPUT TYPE=TEXT NAME="firma" SIZE=30 class=box></td>
</tr>
<tr>
<td><P align=right><b>Ansprechpartner:</b></P></td>
<td><INPUT TYPE=TEXT NAME="ansprech" SIZE=30 class=box></td>
</tr>
<tr>
<td><P align=right><b>Straße:</b></P></td>
<td><INPUT TYPE=TEXT NAME="strasse" SIZE=30 class=box></td>
</tr>
<tr>
<td><P align=right><b>Ort:</b></P></td>
<td><INPUT TYPE=TEXT NAME="ort" SIZE=30 class=box></td>
</tr>
<tr>
<td><P align=right><b>PLZ:</b></P></td>
<td><INPUT TYPE=TEXT NAME="plz" SIZE=30 class=box></td>
</tr>
<tr>
<td colspan=2> </td>
</tr>
<tr>
<td></td>
<td><p>Der Newsletter erscheint ca. alle 6 Wochen.
Sie können den Newsletter selbstverständlich jederzeit wieder
abbestellen.<br><br>
Sämtliche angegebenen Angaben werden ausschließlich zu Ihrer
individuellen Betreuung, der Übersendung von
Produktinformationen oder
der Unterbreitung von Serviceangeboten gespeichert.</p></td>
</tr>
<tr>
```

```
<td colspan=2> </td>
</tr>
<tr>
<td> </td>
<td><input type="image" border="0" name="imageField"
src="images/bestellen.gif"></td>
</tr>
<tr>
<td colspan=2> </td>
</tr>
</table>

</form>
</dtml-if>
```

Listung 3.4: Java Code für Datenabgleich im Auszug

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Iterator;
import java.util.Map;
import java.util.ResourceBundle;
import java.util.TreeMap;

public class ObsImporterApp {
  Connection con = null;

  Map tableList = null;

  public ObsImporterApp() {
    tableList = new TreeMap();
  }

  public void importFromFile(String fileName) {
    int lineNum = 0;
    try
    {
      String prevTableCode = "";
      FileInputStream fis = new FileInputStream(fileName);
      InputStreamReader isr = new InputStreamReader(fis,
"Cp1252");
      System.out.println("open file, use encoding: " +
isr.getEncoding());
      BufferedReader reader = new BufferedReader(isr);
      String line = reader.readLine();
      while (line != null) {
        lineNum++;
        String strTableCode = line.substring(0, 2);
        Table table = (Table) tableList.get(new
Integer(strTableCode));
        if (table == null)
          throw new Exception("unknown table code " +
strTableCode
              + ". Please check ini file!");

        if( !prevTableCode.equals(strTableCode) )
        {
          System.out.println("processing table
'"+table.getName()+"'");
          prevTableCode = strTableCode;
        }

        line = line.substring(2);

        table.parseLine(line);
        table.updateRow();
```

```
        line = reader.readLine();
      } // while
    } catch (FileNotFoundException err) {
      System.out.println("file " + fileName + " not found!");
      return;
    } catch (Exception err) {
      System.out.println("Error reading file " + fileName +
":\n"
          + err.getMessage());
      System.out.println("   line:" + lineNum);
      return;
    }
  }

  public boolean readIniFile(String filename) {
    try {
      BufferedReader reader = new BufferedReader(new
FileReader(filename));
      String line = reader.readLine();
      int lineNum = 0;
      while (line != null) {
        lineNum++;
        line.trim();
        if (line.length() == 0) {
          line = reader.readLine();
          continue;
        }

        if (line.charAt(0) != '[' || !line.endsWith("]")) {
          line = reader.readLine();
          continue;
        }

        String tableName = line.substring(1, line.length() -
1);
        Table table = new Table(tableName, this.con);
        int tableCode = 0;

        line = reader.readLine();
        lineNum++;
        if (line == null) {
          System.out.println("Error in file " + filename + ",
line:" + lineNum);
          System.out.println("   unexpected end of file!");
          return false;
        }
        try {
          tableCode = Integer.parseInt(line);
        } catch (NumberFormatException nfe) {
          System.out.println("Error in file " + filename + ",
line:" + lineNum);
          System.out.println("   Table code (CodTipoRecord)
expected!");
          return false;
        }

        // read columns
        // -----------------------------------------------
        line = reader.readLine();
```

```
        while (line != null && line.length() != 0 &&
line.charAt(0) != '[') {
          lineNum++;
          line.trim();
          if (line.length() != 0) {
            Column col = new Column();
            if (!col.parseIniLine(line)) {
              System.out.println("Error in file " + filename +
", line:"
                  + lineNum);
              return false;
            }
            table.addColumn(col);
          }

          line = reader.readLine();
        } // while columns

        // ------------------------------------------------

        tableList.put(new Integer(tableCode), table);
      } // while tables
    } catch (FileNotFoundException err) {
      System.out.println("readIniFile: file " + filename + "
not found!");
      return false;
    } catch (IOException err) {
      System.out.println("Error reading file " + filename +
":\n"
          + err.getMessage());
      return false;
    }
    return true;
  }

  public void createTables() {
    Iterator iter = tableList.keySet().iterator();
    while (iter.hasNext()) {
      Integer tableCode = (Integer) iter.next();
      System.out.println("CREATE TABLE:  Code: " + tableCode);

      Table table = (Table) tableList.get(tableCode);
      System.out.println(table);

      try {
        table.createInDb();
      } catch (Exception err) {
        System.out.println("Error creating table " +
table.getName() + ":");
        System.out.println(err.getMessage());
      }
    }
  }

  protected void resetUpdateFlags()
  {
    System.out.println("Reset update flags");

    Iterator iter = tableList.values().iterator();
    while( iter.hasNext() )
```

```
    {
      Table table = (Table)iter.next();
      try
      {
        table.resetUpdateFlag();
      }
      catch( Exception err )
      {
        System.err.println("resetUpdateFlags: " + err);
      }
    }
  }

  public boolean openDB() {
    String driver = null;
    try {
      ResourceBundle dbProps = ResourceBundle.getBundle("db");
      driver = dbProps.getString("driver");
      String url = dbProps.getString("url");
      String user = dbProps.getString("user");
      String pass = dbProps.getString("pass");

      Class.forName(driver);

      System.out.println("ObsImporterApp.openDB: trying to
connect to db with:\r\n\t"
                         + dbProps.getString("url") + "
(user:" + user + ")");

      DriverManager.setLogStream(System.out);
      con = DriverManager.getConnection(url, user, pass);

dbMetaData.supportsResultSetConcurrency(ResultSet.TYPE_SCROLL_
INSENSITIVE, ResultSet.CONCUR_UPDATABLE));

      System.out.println("ObsImporterApp.openDB: successfully
connected");
    } catch (ClassNotFoundException e) {
      System.out.println("ObsImporterApp.openDB: db driver " +
driver + " not found.");
      return false;
    } catch (SQLException e) {
      System.out.print("ObsImporterApp.openDB: ");
      System.out.println(e);
      return false;
    }
    return true;
  }

  public void closeDB() {
    try {
      con.close();
      System.out.println("ObsImporterApp.close: successfully
disconnected.");
    } catch (Exception e) {
      System.out.print("ObsImporterApp.closeDB: ");
      System.out.println(e);
    }
  }
```

```
  protected static int findOption(String[] args, String
option)
  {
    option = "-" + option;
    int index = 0;
    while( index < args.length &&
!args[index].equalsIgnoreCase(option))
      index++;
    if( index < args.length )
      return index;
    else
      return -1;
  }

  protected static String getParameter(String[] args)
  {
    int index = 0;
    while( index < args.length && args[index].startsWith("-")
)
      index++;
    if( index < args.length )
      return args[index];
    else
      return null;
  }

  public static void main(String[] args) {
    System.out.println("Starting OBS Importer");
    System.out.println("working dir: " +
System.getProperty("user.dir"));

    if (args.length < 1) {
      System.out.println("Usage: ObsImporterApp <file> [-c|-
r|-d]");
      System.out.println("Options: -c  create tables");
      System.out.println("         -r  reset update flags");
      return;
    }

    ObsImporterApp app = new ObsImporterApp();

    String appName = app.getClass().getName();

    if (app.openDB()) {
      if (!app.readIniFile(appName + ".ini")) {
        System.exit(-1);
      }

      if( findOption(args, "c") >= 0 )
        app.createTables();

      if( findOption(args, "r") >= 0 )
        app.resetUpdateFlags();

      String fileName = getParameter(args);
      if( fileName != null )
        app.importFromFile(fileName);
    }
    app.closeDB();
  }
```

Listung 4.8: Record Log im Auszug

```
[Network Analyzer (1d58: cd8)] ------------------------------------------------
----------------------------------------
[Network Analyzer (1d58: cd8)] Load Network Traffic Analyzers:
[Network Analyzer (1d58: cd8)]     Analyzer Module: WPLUS (value=)
[Network Analyzer (1d58: cd8)]     Analyzer Module: WebBase
(value=GetHttpProtocolAnalyzer:api_http_filter.dll)
[Network Analyzer (1d58: cd8)]     + Network Analyzer: api_http_filter.dll @
GetHttpProtocolAnalyzer Loaded!
[Network Analyzer (1d58: cd8)]     + Interception Auditors:
WinInetWplusInterceptionAudit:api_http_filter.dll
[Network Analyzer (1d58: cd8)]     Analyzer Module: QTWeb (value=)
[Network Analyzer (1d58: cd8)]     Analyzer Module: local_server (value=)
[Network Analyzer (1d58: cd8)] ------------------------------------------------
----------------------------------------
[Network Analyzer (1d58:1610)] Address lookup for TOMBIG = 192.22.11.86
[Network Analyzer (1d58:1ca4)] Address lookup for TOMBIG = 192.22.11.86
[Network Analyzer (1d58:1ca4)] Request Connection: Remote Server @
195.110.210.105:80   (Service=)  (Sid=  1)  PROXIED!
[Web Request      (1d58:1610)] "GET /"
[Network Analyzer (1d58:1610)]   (Sid:  1) Client -> Server : 427 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  1) Server -> Client : 293 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  1) Server -> Client : 315 bytes
(Service=HTTP)
[Network Analyzer (1d58:1ca4)] Address lookup for TOMBIG = 192.22.11.86
[Network Analyzer (1d58:1ca4)] Request Connection: Remote Server @
195.110.210.105:8080   (Service=)  (Sid=  2)  PROXIED!
[Web Request      (1d58:1610)] "GET /pkw/"
[Network Analyzer (1d58:1610)]   (Sid:  2) Client -> Server : 271 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  2) Server -> Client : 293 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  2) Server -> Client : 1070 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  2) Server -> Client : 1521 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  2) Server -> Client : 1366 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  2) Server -> Client : 1494 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  2) Server -> Client : 1426 bytes
(Service=HTTP)
[Network Analyzer (1d58:1ca4)] Address lookup for TOMBIG = 192.22.11.86
[Network Analyzer (1d58:1a48)] Address lookup for TOMBIG = 192.22.11.86
[Network Analyzer (1d58:1610)]   (Sid:  2) Server -> Client : 1283 bytes
(Service=HTTP)
```

```
[Network Analyzer (1d58:1610)]   (Sid:  2) Server -> Client : 1566 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  2) Server -> Client : 1432 bytes
(Service=HTTP)
[Network Analyzer (1d58:1ca4)] Request Connection: Remote Server @
195.110.210.105:8080   (Service=)  (Sid=  3)  PROXIED!
[Web Request      (1d58:1610)] "GET /pkw/stylesheets/fiat.css"
[Network Analyzer (1d58:1610)]   (Sid:  3) Client -> Server : 338 bytes
(Service=HTTP)
[Network Analyzer (1d58:1a48)] Request Connection: Remote Server @
195.110.210.105:8080   (Service=)  (Sid=  4)  PROXIED!
[Web Request      (1d58:1610)] "GET /pkw/stylesheets/start.css"
[Network Analyzer (1d58:1610)]   (Sid:  4) Client -> Server : 339 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  2) Server -> Client : 1430 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  2) Server -> Client : 1414 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  2) Server -> Client : 1284 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  3) Server -> Client : 169 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  3) Server -> Client : 1253 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  3) Server -> Client : 1422 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 169 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 1357 bytes
(Service=HTTP)
[Web Request      (1d58:1610)] "GET /fiatpkw/scripts/js.js"
[Network Analyzer (1d58:1610)]   (Sid:  4) Client -> Server : 331 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  3) Server -> Client : 1422 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  3) Server -> Client : 1142 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 264 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 3326 bytes
(Service=HTTP)
[Web Request      (1d58:1610)] "GET /pkw/images/will-norm.gif"
[Network Analyzer (1d58:1610)]   (Sid:  3) Client -> Server : 338 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  3) Server -> Client : 265 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  3) Server -> Client : 671 bytes
(Service=HTTP)
[Web Request      (1d58:1610)] "GET /pkw/images/habe-norm.gif"
```

```
[Network Analyzer (1d58:1610)]   (Sid:  4) Client -> Server : 338 bytes
(Service=HTTP)
[Web Request      (1d58:1610)] "GET /pkw/images/partner-norm.gif"
[Network Analyzer (1d58:1610)]   (Sid:  3) Client -> Server : 345 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 265 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  3) Server -> Client : 809 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 698 bytes
(Service=HTTP)
[Web Request      (1d58:1610)] "GET /pkw/images/alles-norm.gif"
[Network Analyzer (1d58:1610)]   (Sid:  3) Client -> Server : 339 bytes
(Service=HTTP)
[Web Request      (1d58:1610)] "GET /pkw/images/angebot-norm.gif"
[Network Analyzer (1d58:1610)]   (Sid:  4) Client -> Server : 341 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  3) Server -> Client : 265 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  3) Server -> Client : 610 bytes
(Service=HTTP)
[Web Request      (1d58:1610)] "GET /pkw/images/zubehoer-norm.gif"
[Network Analyzer (1d58:1610)]   (Sid:  3) Client -> Server : 342 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 265 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 502 bytes
(Service=HTTP)
[Web Request      (1d58:1610)] "GET /pkw/images/will-over.gif"
[Network Analyzer (1d58:1610)]   (Sid:  4) Client -> Server : 338 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  3) Server -> Client : 740 bytes
(Service=HTTP)
[Web Request      (1d58:1610)] "GET /pkw/images/habe-over.gif"
[Network Analyzer (1d58:1610)]   (Sid:  3) Client -> Server : 338 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 265 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 644 bytes
(Service=HTTP)
[Web Request      (1d58:1610)] "GET /pkw/images/alles-over.gif"
[Network Analyzer (1d58:1610)]   (Sid:  4) Client -> Server : 339 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  3) Server -> Client : 265 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  3) Server -> Client : 675 bytes
(Service=HTTP)
[Web Request      (1d58:1610)] "GET /pkw/images/fiatpartner-over.gif"
[Network Analyzer (1d58:1610)]   (Sid:  3) Client -> Server : 345 bytes
(Service=HTTP)
```

```
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 265 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 589 bytes
(Service=HTTP)
[Web Request      (1d58:1610)] "GET /pkw/images/angebot-over.gif"
[Network Analyzer (1d58:1610)]   (Sid:  4) Client -> Server : 341 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  3) Server -> Client : 265 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  3) Server -> Client : 526 bytes
(Service=HTTP)
[Web Request      (1d58:1610)] "GET /pkw/images/zubehoer-over.gif"
[Network Analyzer (1d58:1610)]   (Sid:  3) Client -> Server : 342 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 265 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 484 bytes
(Service=HTTP)
[Web Request      (1d58:1610)] "GET /pkw/scripts/nav.js"
[Network Analyzer (1d58:1610)]   (Sid:  4) Client -> Server : 332 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  3) Server -> Client : 265 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  3) Server -> Client : 458 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 264 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 3760 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 4398 bytes
(Service=HTTP)
[Web Request      (1d58:1610)] "GET /pkw/scripts/dqm_loader.js"
[Network Analyzer (1d58:1610)]   (Sid:  3) Client -> Server : 339 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  3) Server -> Client : 281 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  3) Server -> Client : 3393 bytes
(Service=HTTP)
[Web Request      (1d58:1610)] "GET /pkw/scripts/browser_ie.js"
[Network Analyzer (1d58:1610)]   (Sid:  4) Client -> Server : 339 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 281 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 3760 bytes
(Service=HTTP)
[Network Analyzer (1d58:1610)]   (Sid:  4) Server -> Client : 5581 bytes
(Service=HTTP)
[Web Request      (1d58:1610)] "GET /pkw/stylesheets/images/bg_lines.gif"
[Network Analyzer (1d58:1610)]   (Sid:  3) Client -> Server : 349 bytes
(Service=HTTP)
```

Listung 4.9: Script für den Load Test

```
/* ------------------------------------------------------------------
--------------
      Script Title       :
      Script Description :


      Recorder Version   : 1196
   ------------------------------------------------------------------
-------------- */

vuser_init()
{

      web_url("test.webtom.at",
            "URL=http://test.webtom.at/",
            "Resource=0",
            "RecContentType=text/html",
            "Referer=",
            "Snapshot=t1.inf",
            "Mode=HTML",
            EXTRARES,
            "Url=http://195.110.210.105:8080/fiatpkw/images/will-
over.gif", "Referer=http://fiat.webtom.at:8080/fiatpkw/", ENDITEM,
            "Url=http://195.110.210.105:8080/fiatpkw/images/habe-
over.gif", "Referer=http://fiat.webtom.at:8080/fiatpkw/", ENDITEM,
            "Url=http://195.110.210.105:8080/fiatpkw/images/alles-
over.gif", "Referer=http://fiat.webtom.at:8080/fiatpkw/", ENDITEM,

      "Url=http://195.110.210.105:8080/fiatpkw/images/fiatpartner-
over.gif", "Referer=http://fiat.webtom.at:8080/fiatpkw/", ENDITEM,
            "Url=http://195.110.210.105:8080/fiatpkw/images/angebot-
over.gif", "Referer=http://fiat.webtom.at:8080/fiatpkw/", ENDITEM,

      "Url=http://195.110.210.105:8080/fiatpkw/images/zubehoer-
over.gif", "Referer=http://fiat.webtom.at:8080/fiatpkw/", ENDITEM,

      "Url=http://195.110.210.105:8080/fiatpkw/scripts/browser_ie.js
", "Referer=http://fiat.webtom.at:8080/fiatpkw/", ENDITEM,
```

```
      "Url=http://195.110.210.105:8080/fiatpkw/stylesheets/images/bg
_lines.gif", "Referer=http://fiat.webtom.at:8080/fiatpkw/", ENDITEM,
            "Url=http://195.110.210.105:8080/fiatpkw/punto.swf",
"Referer=", ENDITEM,
            "Url=http://195.110.210.105:8080/fiatpkw/logo.jpg",
"Referer=", ENDITEM,

      "Url=http://195.110.210.105:8080/fiatpkw/sounds/sound0.swf",
"Referer=", ENDITEM,

      "Url=http://195.110.210.105:8080/fiatpkw/t0/t0_v_10.swf",
"Referer=", ENDITEM,

      "Url=http://195.110.210.105:8080/fiatpkw/t0/t0_v_31.swf",
"Referer=", ENDITEM,

      "Url=http://195.110.210.105:8080/fiatpkw/t0/t0_v_14.swf",
"Referer=", ENDITEM,

      "Url=http://195.110.210.105:8080/fiatpkw/t0/t0_v_13.swf",
"Referer=", ENDITEM,

      "Url=http://195.110.210.105:8080/fiatpkw/t0/t0_v_12.swf",
"Referer=", ENDITEM,

      "Url=http://195.110.210.105:8080/fiatpkw/stylesheets/images/ba
ckground_left.jpg", "Referer=http://fiat.webtom.at:8080/fiatpkw/",
ENDITEM,

      "Url=http://195.110.210.105:8080/fiatpkw/stylesheets/images/do
ts.png", "Referer=http://fiat.webtom.at:8080/fiatpkw/", ENDITEM,

      "Url=http://195.110.210.105:8080/fiatpkw/t0/t0_v_11.swf",
"Referer=", ENDITEM,

      "Url=http://195.110.210.105:8080/fiatpkw/setLoaded?loaded=1",
"Referer=http://fiat.webtom.at:8080/fiatpkw/", ENDITEM,
```

```
      "Url=http://195.110.210.105:8080/fiatpkw/t0/t0_v_20.swf",
"Referer=", ENDITEM,

      "Url=http://195.110.210.105:8080/fiatpkw/t0/t0_v_29.swf",
"Referer=", ENDITEM,

      "Url=http://195.110.210.105:8080/fiatpkw/t0/t0_v_15.swf",
"Referer=", ENDITEM,

      "Url=http://195.110.210.105:8080/fiatpkw/t0/t0_v_25.swf",
"Referer=", ENDITEM,
            LAST);

      lr_think_time(6);

      web_url("m475",
            "URL=http://195.110.210.105:8080/fiatpkw/will/m475/",
            "Resource=0",
            "RecContentType=text/html",
            "Referer=http://fiat.webtom.at:8080/fiatpkw/",
            "Snapshot=t2.inf",
            "Mode=HTML",
            EXTRARES,
            "Url=images/will-over.gif", ENDITEM,
            "Url=images/habe-over.gif", ENDITEM,
            "Url=images/alles-over.gif", ENDITEM,
            "Url=images/fiatpartner-over.gif", ENDITEM,
            "Url=images/angebot-over.gif", ENDITEM,
            "Url=images/zubehoer-over.gif", ENDITEM,
            "Url=scripts/browser_ie.js", ENDITEM,
            "Url=stylesheets/images/bg_lines.gif", ENDITEM,
            "Url=stylesheets/images/background_left.jpg", ENDITEM,
            "Url=http://www.fiatautomobil.at/fiat/files/pkw-
menu.php?var=:VERSIONEN%20/%20PREISE&red=207&green=205&blue=190",
ENDITEM,
            "Url=http://www.fiatautomobil.at/fiat/files/pkw-
menu.php?var=:DESIGN&red=207&green=205&blue=190", ENDITEM,
            "Url=http://www.fiatautomobil.at/fiat/files/pkw-
menu.php?var=:AUSSTATTUNG&red=207&green=205&blue=190", ENDITEM,
```

```
            "Url=http://www.fiatautomobil.at/fiat/files/pkw-
menu.php?var=:MOTOREN&red=207&green=205&blue=190", ENDITEM,
            "Url=http://www.fiatautomobil.at/fiat/files/pkw-
menu.php?var=:SICHERHEIT&red=207&green=205&blue=190", ENDITEM,
            "Url=http://www.fiatautomobil.at/fiat/files/pkw-
menu.php?var=:CONFIGURATOR&red=207&green=205&blue=190", ENDITEM,
            "Url=setLoaded?loaded=1", ENDITEM,
            LAST);

      lr_think_time(5);

      web_url("ausstattung",

      "URL=http://195.110.210.105:8080/fiatpkw/will/m475/ausstattung
/",
            "Resource=0",
            "RecContentType=text/html",
            "Referer=",
            "Snapshot=t3.inf",
            "Mode=HTML",
            EXTRARES,
            "Url=images/will-over.gif", ENDITEM,
            "Url=images/habe-over.gif", ENDITEM,
            "Url=images/fiatpartner-over.gif", ENDITEM,
            "Url=images/angebot-over.gif", ENDITEM,
            "Url=images/zubehoer-over.gif", ENDITEM,
            "Url=images/alles-over.gif", ENDITEM,
            "Url=scripts/browser_ie.js", ENDITEM,
            "Url=stylesheets/images/bg_lines.gif", ENDITEM,
            "Url=stylesheets/images/background_left.jpg", ENDITEM,
            "Url=http://www.fiatautomobil.at/fiat/files/pkw-
menu.php?var=:VERSIONEN%20/%20PREISE&red=207&green=205&blue=190",
ENDITEM,
            "Url=http://www.fiatautomobil.at/fiat/files/pkw-
menu.php?var=:DESIGN&red=207&green=205&blue=190", ENDITEM,
            "Url=http://www.fiatautomobil.at/fiat/files/pkw-
menu.php?var=:MOTOREN&red=207&green=205&blue=190", ENDITEM,
            "Url=http://www.fiatautomobil.at/fiat/files/pkw-
menu.php?var=:SICHERHEIT&red=207&green=205&blue=190", ENDITEM,
```

```
            "Url=http://www.fiatautomobil.at/fiat/files/pkw-
menu.php?var=:CONFIGURATOR&red=207&green=205&blue=190", ENDITEM,
            "Url=setLoaded?loaded=1", ENDITEM,
            LAST);

      return 0;
}
```

Zusammenstellung der verwendeten Technologien:

Betriebssysteme:

- Suse Linux, Version 10.0
- United Linux, Version 1.1
- Windows XP, SP2

Content Management Systeme:

- Zope 2.7.7-final
- Typo 3
- phpCMS

verwendete Programmiersprachen:

- PHP version 4 und 5
- python 2.4.1
- Java Script
- Java

Weitere Technologien:

- Apache Webserver, Version 2.0.54
- ZServer.HTTPServer.zhttp_server
- ZServer.FTPServer.FTPServer
- MySql 4.1.13
- Postfix 2.2.5
- Webalizer 2.01-10
- Logrotate 3.7.1
- Nagios 2.9